# Revisiting Remote State Preparation with Verifiability: A New Set of Notions with Well-behaved Properties

Jiayu Zhang*[1]

[1]Zhongguancun Laboratory

September 29, 2023

**Abstract**

In remote state preparation with verifiability (RSPV), a client would like to prepare a quantum state (sampled from a state family) on the server side, such that ideally the client knows its full description, while the server holds and only holds the state itself. A closely related notion called self-testing, which is recently generalized to the single-server computationally-secure setting [21], aims at certifying the server's operation. These notions have been widely studied in various different settings and have become fundamental building blocks in many quantum protocols [10, 1, 30, 12]. However, there are many variants of definitions in existing works, and many of these variants do not have some desirable properties like sequential composability. What's more, existing works mainly focus on simple state families like simple product states, and treatments for these types of states are already technically complicated; in this background, a new framework that could potentially support more general solutions is desirable.

In this paper, we choose notions or basic ideas from existing works [3, 10, 30, 28] and introduce new notions, with the goal of developing a more general, well-behaved framework for these problems. We choose RSPV with simulation-based soundness [3, 10, 30] (instead of rigidity-based soundness [1]), and study its basic properties like composability. Furthermore, for controlling the server's operation in a verifiable way, we introduce a new notion named *remote operator application with verifiability* (ROAV) as a replacement of self-testing. In this notion the server is provided with an unknown input state, and is supposed to perform a specific operator (sampled from an operator family) to the state; the client knows the operator description, but what server knows in the end is limited to the output state of the operation applied on the input state. Finally, we show several basic constructions of protocols under our set of notions, and discuss why these notions could potentially lead to quantum cryptographic protocols with new functionalities.

# 1 Introduction

## 1.1 Background

Development of quantum computers leads to demands of various quantum cryptographic protocols, for example, quantum computation verification [19, 30], multiparty quantum computations [2], etc.

*zhangjy@zgc.edu.cn

In its typical setting, there is a client and a remote quantum server (or servers), and the client would like to achieve some quantum cryptographic tasks, but it does not trust the server; thus a cryptographic protocol between the client and the server is needed. Among these problems, two examples that are basic and very important are *remote state preparation* (RSP) [3] and *self-testing* [29], which we introduce below.

### 1.1.1 Remote state preparation

In the RSP problem, ideally, the client would like to prepare a quantum state (sampled from a state family) on the server side; thus in the end the client knows the description of the state, while the server simply holds the state. The trivial solution is to simply send the quantum state through a quantum channel. RSP asks: how could we simulate this quantum communication by other means (like classical communication or other types of quantum communication), possibly under computational assumptions?

Studies of RSP have a long history [25, 3]. One setting [3] of RSP is the fully honest setting: all the parties execute the protocols honestly. In this work, we are interested in the setting where the server could be malicious, and RSP protocols in this setting should satisfy a correctness requirement and a security requirement.

The natural correctness requirement for RSP says that when the server is honest, the client accepts and the server gets the state while the client gets the state description. For security, there are different security notions, including blindness (secrecy) and verifiability (soundness) [7, 10, 31]. In this paper we focus on RSP with verifiability (RSPV). In RSPV, intuitively, the client is able to verify that in case of acceptance the server really gets the state, as if it is sent through a quantum channel. A malicious server who attempts to get other states by deviating from the protocol would be caught cheating by the client.

As a natural quantum task, the RSPV problem is interesting on its own. What's more, it has become an important building block in many other quantum cryptographic protocols. As examples, [10, 7] first construct classical channel cryptography-based RSPV and use it to achieve classical verification of quantum computations; [1] explores more applications of RSPV; [30] takes the RSPV approach to achieve classical verification of quantum computations with linear total time complexity. Many quantum cryptographic protocols rely on quantum channel and quantum communication, and an RSPV protocol could often allow us to replace these quantum communication steps by other cheaper resources, like classical communication.

Preparing states on the server side is quite useful. But in many scenarios what the client needs is to have control on server's *operations*, as introduced below.

### 1.1.2 How to control server's operations

How could the client verify that the server has really applied an operation on its state? In existing works, people raised the notion of self-testing to address the problem.

The concept of self-testing also has a long history [29, 26, 20] in quantum information. One famous application of self-testing is in the study of non-local games [14, 27]. In this scenario, the client (or called verifer) sends questions to two spatially-separated but entangled quantum servers, and quantum servers are supposed to perform specific measurements and send back the results, then the client decides whether to accept or reject. The natural correctness requirement says that when all the parties follow the protocol, the client accepts with some specific probability, say, OPT. Furthermore, specific games have the property that, any servers that want to pass the protocol with probability bigger than OPT $-\epsilon$ have to use a strategy (measurement operators) that is close to the honest behavior. This provides a way to constrain servers' operations through only classical interactions and spatial separation, which is a fundamental technique in the study of non-local games.

Recently a series of works [21, 12, 4, 23] study the single-server analog of two server self-testing as discussed above. The goal is typically to design cryptographic protocols between a client and a single quantum server so that it is certified that the server has prepared the entangled state between two registers as the two server setting, and has performed the measurements on it. [21] studies the basic analog of CHSH game on the single server computationally secure setting and construct a protocol that only uses classical channel; [22, 12] further extend it to the three-qubit and $N$-qubit setting; [16, 23] makes use of QFHE [18] to address the problem; [15, 16, 4] study the proof of quantumness problem and the construction is later proved to have a self-testing property. Typically these self-testing protocols have also achieved a sense of RSPV since the protocols also certify the underlying entangled states; however, these self-testing protocols do not aim to reserve the states in the end.

#### 1.1.3 Subtleties and limitations of existing works

There are several subtleties or limitations in existing works for RSPV or self-testing. First, existing works for RSPV do not have a consistent choices of definitions. There are roughly two types of security notions, the *rigidity-based* (or isometry-based) soundness [7, 1] and *simulation-based* soundness [3, 10, 30]. Roughly speaking, these two definitions go as follows:

- (Rigidity-based soundness) The output state, going through an isometry, is close to the target state.
- (Simulation-based soundness) The target state, going through a simulator, is indistinguishable to the output state.

Existing works do not seem to care about the differences.

Another subtlety in RSPV and self-testing problems is its composability. For example, one basic desirable property of cryptographic primitives is sequential composability between independent instances. This means, if the client and the server execute an RSPV (or self-testing) protocol for a state family $\mathcal{F}_1$, and then execute another protocol for another state family $\mathcal{F}_2$, we would like the overall protocol to be automatically an RSPV for $\mathcal{F}_1 \otimes \mathcal{F}_2$ (defined to be tensor products between each pair of elements). Existing works [1, 12] deal with this type of states or operators by designing new protocols and giving highly technical proofs; if such sequential composability property holds for RSPV or self-testing, protocols for tensor products of simple states could be reduced to protocols for simple states, which will potentially significantly simplify the constructions and proofs.

One more limitation in current RSPV and self-testing protocols is that they could only handle simple tensor product states and operators. Remote preparation of large entangled states is also quite useful in quantum cryptography [8, 5], and a more general solution for RSPV for these types of states is highly desirable. We note that the composability subtlety discussed above also makes the problem harder: considering the fact that preparing simple product states is already highly technically complicated, preparing large entangled states might be too complicated to work on.

In this background, we ask the following question:

*Could RSPV and single-server self-testing be more well-behaved and useful?*

## 1.2 Our Contributions

We argue that the current complicated situation of RSPV and single-server self-testing is largely from the choices of definitions. In this work we choose or introduce a new set of notions for these problems and study their properties and applications, which we summarize below.

### 1.2.1 Choosing or introducing definitions

**RSPV** We first develop a new set of notions. For RSPV, we choose and study RSPV with simulation-based soundness (see Section 1.1.3 and 3.1.2). We show that the definitions that we choose have several desirable properties, which could hopefully make RSPV much easier to work on:

- We show our choice of notions has a well-behaved sequential composability property.
- In usual applications of RSPV, simulation-based definition is as powerful as rigidity-based definition.

Then we introduce a new notion called remote operator application with verifiability (ROAV), as our analog of self-testing in the single-party cryptographic setting.

**Remote operator application with verifiability (ROAV)** Recall in the two-server protocol design scenario, one typical techniques is to design two subprotocols, one of them has a self-testing property, while the other is to execute the computation. One server, without communicating with the other, could not decide which one the client is currently executing; to pass the overall protocol it has to pass the self-testing subprotocol so that its operations has to be close to the honest behavior; and this implies the computation subprotocol is also executed almost honestly. The driven question behind our definition is: could we formulate a notion in the single-server cryptographic setting that is analogous to what a specific server sees in the two-server setting?

We raise the notion of ROAV for formulating this intuition. An ROAV for a target operation $\mathcal{E}$ is defined as a tuple $(\rho_{test}, \pi_{test}, \pi_{comp})$ where:

- $\rho_{test}$ is a specific state used as the input state of $\pi_{test}$.
- $\pi_{comp}$ is a protocol with an undetermined input state whose dimension is the same as the server-side of $\rho_{test}$.

Here $(\rho_{test}, \pi_{test})$ is the test mode, which means, running $\pi_{test}$ on input state $\rho_{test}$ is used to test the adversary's behavior; $\pi_{comp}$ is the computation mode, which means, in this mode the operator $\mathcal{E}$ is finally applied on the input state. More formally, the soundness is defined roughly as follows:

For any adversary $\mathsf{Adv}$, denoting the final output of running protocol $\pi_{...}$ against adversary $\mathsf{Adv}$ on input $\rho_{...}$ as $\pi_{...}^{\mathsf{Adv}}(\rho_{...})$, the ROAV satisfies:

- either the cheating behavior gets caught in $\pi_{test}^{\mathsf{Adv}}(\rho_{test})$ with high probability,
- or $\pi_{comp}^{\mathsf{Adv}}(\chi)$ is close (in a sense) to $\mathcal{E}(\chi)$ where $\chi$ denotes an arbitrary input state.

Finally we note that in the formal notion that we propose we consider a large entangled state which can be collapsed to any $\chi$ by measuring part of its systems.

We argue that our new notion has relatively well-behaved properties, is consistent with the intuition of self-testing in the multi-party setting, and is potentially useful.

### 1.2.2 Applications

We show several potential applications of our notions as follows. First in Section 4.2 we show that ROAV is potentially a useful tool for constructing RSPV protocols for more general state families. The outcome of an ROAV protocol is a remote preparation of joint state $\mathcal{E}(\chi)$ where $\chi$ is the input state; such a state might be hard to prepare directly, but could be made possible once we have an ROAV for $\mathcal{E}$ and have the RSPV for the corresponding $\rho_{test}$ and $\chi$.

Then we construct a Hamiltonian ground energy testing protocol based on specific RSPV and ROAV. This shows the potential of our set of notions in other quantum cryptogrpahic problems like QMA verification. Our construction shares similarities to Grilo's Hamiltonian verification protocol in the 2-party setting [11].

## 1.3 More Related Works

One work that shares similarities to our work is [28]. This work studies the complexity of interactive synthesis of states and unitaries. In a sense, the relation of states and unitaries in their work is similar to the relation of RSPV and ROAV in our work; but state complexity problem and the RSPV/ROAV problem seem quite different and have different applications.

## 1.4 Open Questions and Summary

The obvious open question coming out of this work is to give a construction for ROAV. Our work focuses on the definitions and applications in an abstract sense; an explicit construction of ROAV would allow us to instantiate these applications.

We hope our work clarifies the subtleties in RSPV and related problems and could serve as a foundation for further studies.

# Acknowledgements

This work is supported by different fundings at different time during its preparation:

- Partially supported by the IQIM, an NSF Physics Frontiers Center (NSF Grant PHY-1125565) with support of the Gordon and Betty Moore Foundation (GBMF-12500028).
- This work is partially done when the author was visiting Simons Institute for Theory of Computing.
- This work is partially done in Zhongguancun Laboratory.

# 2 Preliminaries

We refer to [24] for basics of quantum computing, and refer to [17] for basics of cryptography. In this section we clarify some notations and notions.

**Notation 2.1.** We use $[m]$ to denote $\{1, 2, \cdots m\}$.

**Notation 2.2.** We use $D(\mathcal{H})$ to denote the set of density operators over some Hilbert space $\mathcal{H}$.

**Notation 2.3.** For a pure state $|\Phi\rangle$, $\Phi$ is an abbreviation of $|\Phi\rangle\langle\Phi|$.

**Notation 2.4.** We use $\mathcal{E}(\rho)$ to denote the operation of an operator (either unitary or superoperator) on density operator $\rho$. We also use this notation when $\mathcal{E}$ is an isometry (say, $V$): it is the same as $V\rho V^\dagger$.

When the system that $\mathcal{E}$ is contained in the system of $\rho$, the operation on the remaining system is identity.

**Notation 2.5.** We use $\rho \approx_\epsilon \sigma$ to denote $|\rho - \sigma|_{\mathrm{tr}} \leq \epsilon$, where $|\cdot|_{\mathrm{tr}}$ is the trace distance.

**Definition 2.1** (Bell basis)**.** In a two qubit system, define the following four states as the Bell basis:

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle),$$

$$\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle).$$

Define $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, these states could be denoted as $\mathsf{X}^a\mathsf{Z}^b|\Phi\rangle$, where $\mathsf{X}^a$ means apply $\mathsf{X}$ if $a = 1$ and apply identity if $a = 0$. $\mathsf{Z}^b$ is defined similarly.

Now define the Bell-basis measurement as follows: the projection onto Bell basis $\mathsf{X}^a\mathsf{Z}^b|Phi\rangle$ has output $(a, b)$.

**Definition 2.2.** In cryptographic protocols there is usually a completeness requirement and a soundness requirement. When both requirements are probabilistic, the statements are stated in the following type:

- (Completeness) In the yes-instance the honest server makes the client accepts with probability $c$.
- (Soundness) In the no-instance the malicious server could at most make the client accepts with probability $s$.

There should be $0 < s < c < 1$. $1 - c$ is called the completeness error. $s$ is called soundness or soundness error.

**Notation 2.6.** In this paper we use $\pi$ to denote cryptographic protocols. Cryptographic protocols typically will take a security parameter as part of inputs; in this paper we denote it as $\kappa$. When we analyze security of protocols, operators and states are typically families of operators or states parameterized by $\kappa$; in this paper we make it implicit.

In the end the protocol will also output a $flag \in \{\mathsf{pass}, \mathsf{fail}\}$; in this work this decision will be made solely on the client side and we denote the projection onto the passing space as $\Pi_{\mathsf{pass}}$.

We use $\pi^{\mathsf{Adv}}(\rho_{in})$ to denote the output joint state of $\pi$ run on initial state $\rho_{in}$ against adversary $\mathsf{Adv}$.

**Notation 2.7.** We write $\rho \approx_\epsilon^{ind:\mathcal{F}} \sigma$ when $\forall \mathsf{Adv} \in \mathcal{F}, \Pr[\mathsf{Adv}(\rho) \to 1] \approx_{\epsilon + \mathsf{negl}(\kappa)} \Pr[\mathsf{Adv}(\sigma) \to 1]$. We write $\rho \approx_\epsilon^{ind} \sigma$ when $\mathcal{F}$ is taken to be all the polynomial time algorithms.

**Fact 1.** *If $\rho \approx_\epsilon^{ind} \sigma$, $\mathcal{E}$ is an efficient operator, then $\Pi_{\mathsf{pass}}(\mathcal{E}(\rho)) \approx_\epsilon^{ind} \Pi_{\mathsf{pass}}(\mathcal{E}(\sigma))$.*

**Fact 2** (Chernoff bounds)**.** *Suppose for all $i \in [K]$, $s_i$ is a random variable independently sampled from $\{0,1\}$ with probability $1-p, p$ corresponding to values $0, 1$. Then*

$$\Pr[\sum_{i\in[K]} s_i \geq (1+\delta)pK] \leq e^{-\delta^2 pK/3}$$

Finally we review the local Hamiltonian problem.

**Definition 2.3** ([11])**.** The following problem is called the XZ k-local Hamiltonian problem:

Given input $(H, a, b)$ where $H$ is a Hamiltonian on $n$-qubit registers, $a, b$ are real value function of $n$, and they satisfy:

$$H = \sum_{j\in[m]} \gamma_j H_j, \quad \forall j, |\gamma_j| \leq 1 \tag{1}$$

$$\forall j, H_j \in \{\sigma_X, \sigma_Z, I\}^{\otimes n} \text{ with at most } k \text{ appearances of non-identity terms} \tag{2}$$

Decide which is the case:

- Yes-instance: The ground energy of $H$ is $\leq a$
- No-instance: The ground energy of $H$ is $\geq b$.

**Theorem 2.1** ([13])**.** *There exist $a(n), b(n) \in [0,1], b - a \geq 1/\mathsf{poly}(n)$ such that the XZ 5-local Hamiltonian problem is QMA-complete.*

# 3 Remote State Preparation with Verifiability

In this section we study definitions and basic properties of RSPV.

## 3.1 Definitions

Recall that in RSPV, the client aims at creating a state sampled from a state ensemble on the server-side. The client should know the description of the state, while the server holds the state itself. Similar to many cryptographic problems, an RSPV protocol needs to have completeness (correctness) and soundness (verifiability). Furthermore, there are two definitions for the soundness of RSPV: simulation-based soundness [3, 10, 30] and rigidity-based soundness [7, 10, 1], which are both used in existing works. In this subsection we choose formal definitions for both variants and study their differences and relations.

### 3.1.1 Basic settings and completeness

To formalize the completeness and soundness of this notion, let's first formalize some basic settings of RSPV. In more detail, we define the target state of RSPV, as follows:

**Definition 3.1.** An RSPV protocol is defined with respect to an ensemble of normalized states and the corresponding probabilities

$$((p_1, |\varphi_1\rangle), (p_2, |\varphi_2\rangle), \cdots (p_D, |\varphi_D\rangle)), \sum_{i\in[D]} p_i = 1.$$

The target state of an RSPV protocol is denoted by the following joint state of the client and the server (described in terms of density operators):

$$\rho_{tar} = \sum_{i\in[D]} p_i \underbrace{|i\rangle\langle i|}_{\text{client}} \otimes \underbrace{|\varphi_i\rangle\langle\varphi_i|}_{\text{server}} \tag{3}$$

And we simply call it RSPV for $(|\varphi_1\rangle \cdots |\varphi_D\rangle)$ when $(p_i)_{i\in[D]}$ is a uniform distribution.

Note that (3) should be intuitively understood as a cq-state; the fact that the client-side register is classical is equivalent to say any operator (for example, distinguishers that will be used later) that operates on the client-side register in (3) only has classical access to it.

Then the completeness of an RSPV protocol is defined as follows.

**Definition 3.2** (Completeness of RSPV)**.** We say an RSPV protocol for target state $\rho_{tar}$ has completeness error $\gamma$ if in the honest setting, in the end of the protocol the joint state of the client and the server is $\gamma$-close to $\rho_{tar}$ (together with the passing flag). And we simply say the protocol is complete if it has completeness error $\mathsf{negl}(\kappa)$.

### 3.1.2 Rigidity-based soundness and simulation-based soundness

The soundness of RSPV is more subtle; below we formalize and study the two types of soundness definitions.

**Rigidity-based soundness** Roughly speaking, the rigidity-based soundness says the output state, after going through an isometry on the server side, is close to the target state. An interpretation is "the passing flag certifies that the server has really got the state".

**Definition 3.3** (Rigidity-based soundness for RSPV)**.** We say a protocol $\pi$ is an RSPV for target state $\rho_{tar}$ with soundness error $\delta$ and approximation error $\epsilon$ under rigidity-based definition if:

For any BQP adversary $\mathsf{Adv}$, any input state $\rho_{in} \in D(\mathcal{S}\otimes\mathcal{T})$ prepared by the adversary where $\mathcal{S}$ is the server-side system and $\mathcal{T}$ is a system that will not be touched by any party in the protocol, there exists a server-side efficiently-computable isometry $V^{\mathsf{Adv}}$ and an efficiently-computable operation $\mathsf{Sim}^{\mathsf{Adv}}$ operated on $\mathcal{S}$ such that:

- (Small passing probability) Either:

$$\text{tr}(\Pi_{\mathsf{pass}}(\pi^{\mathsf{Adv}}(\rho_{in}))) \leq \delta,$$

- or:

$$\Pi_{\mathsf{pass}}(V^{\mathsf{Adv}}(\pi^{\mathsf{Adv}}(\rho_{in}))) \approx^{ind}_{\epsilon} \Pi_{\mathsf{pass}}(\rho_{tar} \otimes \mathsf{Sim}^{\mathsf{Adv}}(\rho_{in})) \tag{4}$$

  where the distinguisher has classical access to the client side of $\rho_{tar}$ and quantum access to all the other registers (including $\mathcal{S}$ and $\mathcal{T}$).

We note that this definition is slightly different from the (rigidity-based) definitions in existing works [10, 1]. In [10, 1] the left hand side of (4) is statistically close to a state in the form of $\sum_i |\varphi_i\rangle\langle\varphi_i| \otimes \sigma_i$, and then a computaitonal indistinguishability requirement is put on $\sigma_i$ for different $i$. We argue that our global indistinguishability captures the same intuition and is more general; what's more, a suitable formulation of variants of definitions in [10, 1] should imply this definition.[1]

[1] One obstacle of showing a direct implication from the definitions in [10, 1] is on the efficient preparable property

**Simulation-based soundness** Different from the rigidity-based soundness, the simulation-based soundness does not certify that the server really holds the state; an interpretation is "the passing flag certifies that what the adversarial server gets is no more than holding the state". It's not as strong as the rigidity-based definition on its own, but arguably it's sufficiently strong for many applications and it turns out to have good properties.

**Definition 3.4** (Simulation-based soundness for RSPV)**.** We say a protocol $\pi$ is an RSPV for target state $\rho_{tar}$ with soundness error $\delta$ and approximation error $\epsilon$ under simulation-based definition if:

For any BQP adversary $\mathsf{Adv}$, any input state $\rho_{in} \in D(\mathcal{S} \otimes \mathcal{T})$ prepared by the adversary where $\mathcal{S}$ is the server-side system and $\mathcal{T}$ is a system that will not be touched by any party in the protocol, there exists an efficiently-computable operation $\mathsf{Sim}^{\mathsf{Adv}}$ operated on $\mathcal{S}$ such that:

- (Small passing probability) Either:

$$\mathrm{tr}(\Pi_{\mathsf{pass}}(\pi^{\mathsf{Adv}}(\rho_{in}))) \leq \delta,$$

- or:

$$\Pi_{\mathsf{pass}}(\pi^{\mathsf{Adv}}(\rho_{in})) \approx^{ind}_{\epsilon} \Pi_{\mathsf{pass}}(\mathsf{Sim}^{\mathsf{Adv}}(\rho_{tar} \otimes \rho_{in})) \tag{5}$$

  where the distinguisher has classical access to the client side of $\rho_{tar}$ and quantum access to all the other registers (including $\mathcal{S}$ and $\mathcal{T}$).

We note that in both notions, we consider initial states that are possibly correlated or entangled between the server's system $\mathcal{S}$ and the running environment $\mathcal{T}$ of cryptographic protocols. This part could be used to model everything else that happens outside this protocol and helps to give RSPV the sequential composability property (and hopefully other types of composability).

Finally, we could prove the simulation-based soundness as defined above is no stronger than the rigidity-based soundness defined above:

**Theorem 3.1.** *Suppose $\pi$ is an RSPV for target state $\rho_{tar}$ with soundness error $\delta$ and approximation error $\epsilon$ under rigidity-based soundness, it's also an RSPV with the same configurations under simulation-based soundness.*

*Proof.* By the rigidity-based soundness we get $V$, $\mathsf{Sim}$ that satisfies (4). Then taking

$$\mathsf{Sim}'(\underbrace{\cdot}_{\rho_{tar}} \otimes \underbrace{\cdot}_{\rho_{in}}) = V^{\dagger}(\underbrace{\cdot}_{\rho_{tar}} \otimes \mathsf{Sim}(\underbrace{\cdot}_{\rho_{in}}))$$

as the simulator in (5) completes the proof. □

But the inverse is not necessarily true. Actually, the rigidity-based soundness of RSPV is not even resilient to an additional empty timestep (that is, no party does anything) at the end of the protocol: the adversary could destroy everything in the end to violate the rigidity requirement. For comparison, the simulation-based notion has such resilience: the state destroying operation could be absorbed into the simulator in (5). However, arguably this also means the simulation-based notion has more well-behaved properties.

on these $\sigma_i$: the definitions of rigidity-based soundness used in [10, 1] does not seem to imply it could be written in the form of $\mathsf{Sim}(\rho_{in})$. A more careful analysis of the relations between this definition and existing rigidity definitions remains to be done and is out of the scope of this work.

What's more, intuitively the simulation-based version is as useful as the rigidity-based version in common applications of RSPV. When we construct cryptographic protocols, what we are doing is usually to enforce that the malicious parties could not do something. In this sense, in the simulation-based soundness it is certified that what the adversary gets is no more than the target state, which should be at least as secure as really getting the target state.

## 3.2 Basic Properties of RSPV with Simulation-based Soundness

Below we prove several useful properties of simulation-based RSPV.

### 3.2.1 Composition property

First we could prove the simulation-based RSPV has a natural sequential composition property. As far as we know, rigidity-based RSPV does not seem to behave well under this property.

**Theorem 3.2** (Sequential composition of RSPV). *Under simulation-based notion, if $\pi_1$ is an RSPV protocol for $\rho_{tar}$ with soundness $s$ and approximation error $\epsilon_1$, $\pi_2$ is an RSPV protocol for $\sigma_{tar}$ with soundness $s$ and approximation error $\epsilon_2$, the honest behavior of $\pi_1$ and $\pi_2$ are completely independent, then $\pi_2 \circ \pi_1$ is an RSPV protocol for $\rho_{tar} \otimes \sigma_{tar}$ with soundness $s$ and approximation error $\epsilon_1 + \epsilon_2$.*

*Proof.* For an adversary, suppose the initial joint state is $\rho_0 \in D(\mathcal{S} \otimes \mathcal{T})$, the output state of $\pi_1$ with $\rho_0$ being the initial state is $\rho_1 \in D(\mathcal{S} \otimes \mathcal{T})$, and the final output state of $\pi_2$ with $\rho_1$ being the initial state is $\rho_2 \in D(\mathcal{S} \otimes \mathcal{T})$. Then by the simulation-based soundness of $\pi_2$ there exists an efficiently computable simulator $\mathsf{Sim}_2$ working on $\mathcal{S}$ such that:

$$\Pi_{\mathsf{pass}}(\rho_2) \approx^{ind}_{\epsilon_2} \Pi_{\mathsf{pass}}(\mathsf{Sim}_2(\sigma_{tar} \otimes \rho_1)) \tag{6}$$

By the simulation-based soundness of $\pi_1$ there exists an efficiently computable simulator $\mathsf{Sim}_1$ such that:

$$\Pi_{\mathsf{pass}}(\rho_1) \approx^{ind}_{\epsilon_1} \Pi_{\mathsf{pass}}(\mathsf{Sim}_1(\rho_{tar} \otimes \rho_0)) \tag{7}$$

which by Fact 1 implies

$$\Pi_{\mathsf{pass}}(\mathsf{Sim}_2(\sigma_{tar} \otimes \rho_1)) \approx^{ind}_{\epsilon_1} \Pi_{\mathsf{pass}}(\mathsf{Sim}_2(\sigma_{tar} \otimes \mathsf{Sim}_1(\rho_{tar} \otimes \rho_0))) \tag{8}$$

Combining (6)(8) and choosing

$$\mathsf{Sim}(\sigma_{tar} \otimes \rho_{tar} \otimes \cdot) := \mathsf{Sim}_2(\sigma_{tar} \otimes \mathsf{Sim}_1(\rho_{tar} \otimes \cdot))$$

as the final simulator completes the proof. □

### 3.2.2 Cut-and-choose soundness amplification procedure

Consider an RSPV protocol with soundness $s$. We want $s$ to be small. However, very frequently, in some initial construction of RSPV, $s$ might be not good enough (for example, $s$ might be very close to 1). In this case, a soundness amplification procedure is needed.

One commonly used technique for soundness amplification is the *cut-and-choose*. In this technique, to amplify an RSPV protocol $\pi$ with soundness $s$, both parties run many repetitions of $\pi$, and it's required that the server should pass in all the subprotocols. Intuitively if the server wants to pass the overall protocol with high probability, the number of iterations that it could cheat will be relatively small (recall that a cheating server in a single execution of $\pi$ is caught with probability $1-s$). Then a state (and its corresponding classical description) is randomly chosen from these output states.

**Protocol 1** (Cut-and-choose for RSPV)**.** *Given an RSPV protocol $\pi$ for target state $\rho_{tar}$ and a repetition number $L$. The cut-and-choose amplification procedure is defined as below.*

1. *For each $i \in [L]$:*

   (a) *Run $\pi$. Both parties keep the state. The client rejects if $\pi$ rejects.*

2. *The client randomly chooses $i \in [L]$ and sends it to the server. Both parties use the output from the $i$-th repetition as the output state.*

We have the following theorem on this cut-and-choose process. Note this process does not reduce the approximation error but make the soundness better.

**Theorem 3.3.** *If $\pi$ is an RSPV with soundness $s$ and approximation error $\epsilon$, for any $s' < s$, Protocol 1 has soundness $s'$ and approximation error $\epsilon + \frac{2}{L}\log_s(s')$.*

Especially, by taking $L = O(\frac{1}{\epsilon(1-s)})$, we are able to amplify the original protocol to a new protocol with a much smaller soundness value, and approximation error $O(\epsilon)$.

*Proof.* Consider an adversary $\mathsf{Adv}$. Define event $E_i$ = "the adversary passes by the $i$-th iteration". Then by the simulation-based soundness property we get, for any $i$, there exists an efficiently computable simulator $\mathsf{Sim}_i$ such that either $\Pr[E_i|E_{i-1}] < s$, or (5) is satisfied by the end of the $i$-th iteration.

Suppose this adversary could pass the protocol with overall probability $\geq s'$. Define $S_{\text{low pass}}$ as the set of $i$ that satisfies $\Pr[E_i|E_{i-1}] < s$. To pass the overall protocol the adversary needs to pass in each iteration, thus to pass the overall protocol with probability $\geq s'$, there has to be $|S_{\text{low pass}}| \leq \log_s(s')$.

Denote the initial state as $\rho_0$, and denote the output state by the end of the $i$-th round as $\rho_i$. Then for each $i \in [L] - S_{\text{low pass}}$,

$$\Pi_{\mathsf{pass}}(\rho_i) \approx^{ind}_{\epsilon} \Pi_{\mathsf{pass}}(\mathsf{Sim}_i(\rho_{tar} \otimes \rho_{i-1}))$$

which implies

$$\Pi_{\mathsf{pass}}(\pi_{>i}(\rho_i)) \approx^{ind}_{\epsilon} \Pi_{\mathsf{pass}}(\pi_{>i}(\mathsf{Sim}_i(\rho_{tar} \otimes \pi_{<i}(\rho_0)))) \tag{9}$$

where $\pi_{>i}$ is the protocol after round $i$, and $\pi_{<i}$ is the protocol before round $i$.

In the second round the client makes a random choice of $i \in [L]$. We will construct a simulator that simulates the overall state. The simulator $\mathsf{Sim}$ applied on $(\rho_{tar} \otimes \rho_0)$ is defined as follows:

1. Sample a random coin $i \leftarrow [L]$.

2. Run $\tilde{\pi}_{<i}$ on $\rho_0$ and get $\tilde{\rho}_{i-1}$.

3. Run $\mathsf{Sim}_i$ on $\rho_{tar} \otimes \tilde{\rho}_{i-1}$.

4. Run $\tilde{\pi}_{>i}$ on $\mathsf{Sim}_i(\rho_{tar} \otimes \tilde{\rho}_{i-1})$.

where $\tilde{\pi}$ denotes the simulated protocol execution of $\pi$: instead of interacting with the client, the simulator does all the client-side operations on its own registers and disgards these registers in the end.

We prove this simulator achieves what we want.

use $\mathsf{Disgard}[\cdots]$ to denote the operation of disgarding the client-side registers with specific indices, which is in the second step of Protocol 1. Then by (9) we have

$$\Pi_{i\in[L]-S_{\text{low pass}}}(\sum_{i\in[L]}\frac{1}{L}\left|i\right\rangle\left\langle i\right|\otimes\mathsf{Disgard}[[L]-i](\Pi_{\mathsf{pass}}(\pi_{>i}(\rho_i)))) \tag{10}$$

$$\approx^{ind}_{\epsilon}\Pi_{i\in[L]-S_{\text{low pass}}}(\sum_{i\in[L]}\frac{1}{L}\left|i\right\rangle\left\langle i\right|\otimes\mathsf{Disgard}[[L]-i](\Pi_{\mathsf{pass}}(\pi_{>i}(\mathsf{Sim}_i(\rho_{tar}\otimes\pi_{<i}(\rho_0)))))) \tag{11}$$

By $|S_{\text{low pass}}|\leq\log_s(s')$ there is

$$\Pi_{i\in[L]-S_{\text{low pass}}}(\sum_{i\in[L]}\frac{1}{L}\left|i\right\rangle\left\langle i\right|\otimes\mathsf{Disgard}[[L]-i](\Pi_{\mathsf{pass}}(\pi_{>i}(\rho_i)))) \tag{12}$$

$$\approx_{\frac{1}{L}\log_s(s')}\sum_{i\in[L]}\frac{1}{L}\left|i\right\rangle\left\langle i\right|\otimes\mathsf{Disgard}[[L]-i](\Pi_{\mathsf{pass}}(\pi_{>i}(\rho_i))) \tag{13}$$

$$\Pi_{i\in[L]-S_{\text{low pass}}}(\sum_{i\in[L]}\frac{1}{L}\left|i\right\rangle\left\langle i\right|\otimes\mathsf{Disgard}[[L]-i](\Pi_{\mathsf{pass}}(\pi_{>i}(\mathsf{Sim}_i(\rho_{tar}\otimes\pi_{<i}(\rho_0)))))) \tag{14}$$

$$\approx_{\frac{1}{L}\log_s(s')}\sum_{i\in[L]}\frac{1}{L}\left|i\right\rangle\left\langle i\right|\otimes\mathsf{Disgard}[[L]-i](\Pi_{\mathsf{pass}}(\pi_{>i}(\mathsf{Sim}_i(\rho_{tar}\otimes\pi_{<i}(\rho_0))))) \tag{15}$$

Combining them completes the proof. □

# 4 Remote Operator Application with Verifiability

In this section we introduce a new notion named *remote operator application with verifiability* (ROAV), for certifying server's operations. We will give the definition, and show how to use this notion to construct other RSPV protocols and the energy test protocol.

## 4.1 Definitions of ROAV

**Definition 4.1.** An ROAV for a tuple of operators $(E_1, E_2\cdots E_D)$ is in the form of $(\rho_{test}, \pi_{test}, \pi_{comp})$ where:

- $\rho_{test}$ is in the form of (3) Definition 3.1; $\pi_{test}, \pi_{comp}$ are protocols as defined in Notation 2.6; there is a specific register on the server-side, and the honest behavior of both $\pi_{test}$ and $\pi_{comp}$ take this register as part of their inputs, and:
    - the server-side of $\rho_{test}$ is expected to be in this register in the execution of $\pi_{test}$;

  - the input of $\pi_{comp}$ on this register is not expected to be a specific state; when we describe it (together with the corresponding client-side information) below, we typically use symbol $\chi$.

- $(E_1, E_2 \cdots E_D)$ is a tuple of operators operating on a server-side register and they satisfy $\sum_{i\in[D]} E_i^\dagger E_i = I$.

Here $(E_1, E_2 \cdots E_D)$ are the operators to be verified. Similar to Definition 3.1, we define the target operator as the following superoperator on both the client side and the server side, working on the server-side register and producing outputs on both the server-side register and a client-side register:[2]

$$\mathcal{E}(\underbrace{\cdot}_{\text{server}}) = \sum_{i\in[D]} \underbrace{|i\rangle\langle i|}_{\text{client}} \otimes \underbrace{E_i(\cdot)E_i^\dagger}_{\text{server}}$$

An informal description of our ROAV notion is as follows. In Definition 4.1 $(\rho_{test}, \pi_{test})$ is used to certify the server's operation. Explicitly, suppose the adversary is $\mathsf{Adv}$, then protocol $\pi_{test}^{\mathsf{Adv}}(\rho_{test})$ is used to certify the server's operation. Our goal is to certify that the server has applied the operator $\mathcal{E}$ on the server-side input state, which means, $\pi_{comp}^{\mathsf{Adv}}(\cdot)$ is close to $\mathcal{E}(\cdot)$ where we use $\cdot$ to denote an arbitrary input state. We further note that $\mathsf{Adv}$ is the same adversary in both possible running above, and whether the client is running $\pi_{test}$ or $\pi_{comp}$ is not revealed in advanced.

The completeness and soundness are defined as follows.

**Definition 4.2** (Completeness of ROAV)**.** $(\rho_{test}, \pi_{test}, \pi_{comp})$ is an ROAV for target operator $\mathcal{E}$ with completeness error $\gamma$ if in the honest setting, for any input state $\chi$, the joint output state of the client and the server is $\gamma$-close to $\mathcal{E}(\chi)$. We simply say the protocol is complete if $\gamma = \mathsf{negl}(\kappa)$.

The soundness is formulated by a simulation-based definition. One additional subtlety is whether the server-side registers of $\mathcal{E}$ is contained in or could be bigger than the server-side of $\chi$. We will first formulate the simpler case where the server-side of $\mathcal{E}$ is contained in $\chi$; then we formulate the more general case.

### 4.1.1 Simpler case: the server-side of $\mathcal{E}$ is contained in the server-side register of $\chi$

**Definition 4.3** (Soundness of ROAV)**.** $(\rho_{test}, \pi_{test}, \pi_{comp})$ is an ROAV for target operator $\mathcal{E}$ with soundness error $\delta$ and approximation error $\epsilon$ if: For any BQP adversary $\mathsf{Adv}$, there exists an efficiently computable simulator $\mathsf{Sim}^{\mathsf{Adv}}$ such that for any state $\rho_{in} \in D(\mathcal{S} \otimes \mathcal{T})$ prepared by the adversary where $\mathcal{S}$ is the server-side system and $\mathcal{T}$ is a system that will not be touched by any party in the protocol, one of the following two is true:

- (Small passing probability) when $\rho$ is taken to be $\rho_{test}$:

$$\mathrm{tr}(\Pi_{\mathsf{pass}}(\pi_{test}^{\mathsf{Adv}}(\rho_{test} \otimes \rho_{in}))) \leq \delta$$

- Define

$$|\Phi\rangle = \frac{1}{\sqrt{D}} \sum_{i\in[D]} \underbrace{|i\rangle}_{\text{client}} \otimes \underbrace{|i\rangle}_{\text{server}} \tag{16}$$

[2] Note that we are not using Notation 2.4 for $E_i(\cdot)E_i^\dagger$ to be consistent with the usual notations.

then there is

$$\Pi_{\mathsf{pass}}(\pi_{comp}^{\mathsf{Adv}}(\Phi \otimes \rho_{in})) \approx_{\epsilon}^{ind} \Pi_{\mathsf{pass}}(\mathsf{Sim}^{\mathsf{Adv}}(\mathcal{E}(\Phi) \otimes \rho_{in})) \tag{17}$$

where the distinguisher has classical access to the client side output of $\mathcal{E}$ and quantum access to all the registers (including the client-side of $\Phi$, system $\mathcal{S}$, and $\mathcal{T}$).

#### 4.1.2 General definition of ROAV soundness

Here we further generalize the notion to the setting where $\mathcal{E}$ might operate on a server-side register that is possibly bigger than the server-side of $\rho_{test}$. The soundness definition is mostly the same as Definition 4.3, with differences on (17) and an additional simulator.

**Definition 4.4** (Soundness of ROAV)**.** $(\rho_{test}, \pi_{test}, \pi_{comp})$ is an ROAV for target operator $\mathcal{E}$ with soundness error $\delta$ and approximation error $\epsilon$ if: For any BQP adversary $\mathsf{Adv}$, there exist efficiently computable simulators $\mathsf{Sim}^{\mathsf{Adv}}$ and $\mathsf{Sim}_{in}^{\mathsf{Adv}}$ such that for any state $\rho_{in} \in D(\mathcal{S} \otimes \mathcal{T})$ prepared by the adversary where $\mathcal{S}$ is the server-side system and $\mathcal{T}$ is a system that will not be touched by any party in the protocol, one of the following two is true:

- (Small passing probability) when $\rho$ is taken to be $\rho_{test}$:

$$\mathrm{tr}(\Pi_{\mathsf{pass}}(\pi_{test}^{\mathsf{Adv}}(\rho_{test} \otimes \rho_{in}))) \leq \delta$$

- Define

$$|\Phi\rangle = \frac{1}{\sqrt{D}} \sum_{i \in [D]} \underbrace{|i\rangle}_{\text{client}} \otimes \underbrace{|i\rangle}_{\text{server}} \tag{18}$$

  then there is

$$\Pi_{\mathsf{pass}}(\pi_{comp}^{\mathsf{Adv}}(\Phi \otimes \rho_{in})) \approx_{\epsilon}^{ind} \Pi_{\mathsf{pass}}(\mathsf{Sim}^{\mathsf{Adv}}(\mathcal{E}(\Phi \otimes \mathsf{Sim}_{in}^{\mathsf{Adv}}(\rho_{in})))) \tag{19}$$

  where the distinguisher has classical access to the client-side outputs of $\mathcal{E}$ and quantum access to all the registers.

With this definition, we could handle the case where some server-side states are not known by the client. For example, if the client wants to force the server to apply an operation on a QMA witness state, this definition will be needed.

#### 4.1.3 Basic properties

We show that, under our definition, ROAV has a relatively well-behaved property which allows us to derive ROAV for larger operators in the form of tensor products from ROAV for simpler operators.

**Theorem 4.1.** *Suppose $(\rho_{test,1}, \pi_{test,1}, \pi_{comp,1})$ is an ROAV under simpler definition (Definition 4.3) for target operator $\mathcal{E}_1$ soundness error $\delta$ and approximation error $\epsilon_1$, $(\rho_{test,2}, \pi_{test,2}, \pi_{comp,2})$ is an ROAV (also under Definition 4.3) for target operator $\mathcal{E}_2$ soundness error $\delta$ and approximation error $\epsilon_2$, $\mathcal{E}_1$ and $\mathcal{E}_2$ operate on different registers, the server-side dimension of $\rho_{test,1}$ is $D_1$ and the server-side dimension of $\rho_{test,2}$ is $D_2$.*

*Consider the protocol $(\rho_{test}, \pi_{test}, \pi_{comp})$ where:*

- $$\rho_{test} = \frac{1}{2} \underbrace{|1\rangle\langle 1|}_{\textit{client side register of roundtype}} \otimes \rho_{test,1} \otimes \frac{1}{D_2}\mathbb{I} + \frac{1}{2}|2\rangle\langle 2| \otimes \frac{1}{D_1}\mathbb{I} \otimes \rho_{test,2}$$

- *$\pi_{test}$ is defined as follows. The client chooses to execute one of the following depending on the value of roundtype register, without telling the server the value of roundtype:*
  - *If roundtype $= 1$, execute $\pi_{test,1}$.*
  - *If roundtype $= 2$, execute $\pi_{test,2} \circ \pi_{comp,1}$*

- $$\pi_{comp} := \pi_{comp,2} \circ \pi_{comp,1}$$

*Then $(\rho_{test}, \pi_{test}, \pi_{comp})$ is an ROAV for target operator $\mathcal{E}_1 \otimes \mathcal{E}_2$ with soundness error $\delta' = 1 - \frac{1}{2}(1 - \delta) + \frac{1}{2}\epsilon_1$ and approximation error $\epsilon' = \epsilon_1 + \epsilon_2$.*

We note that our composition protocol could only handle the case where the ROAVs are under the simpler definition (Definition 4.3).

*Proof.* Suppose an adversary $\mathsf{Adv}$ satisfies

$$\mathrm{tr}(\Pi_{\mathsf{pass}}(\pi_{test}^{\mathsf{Adv}}(\rho_{test} \otimes \rho_{in}))) > \delta' \tag{20}$$

Then by the construction of $\pi_{test}$ considering roundtype $= 1$ there is

$$\mathrm{tr}(\Pi_{\mathsf{pass}}(\pi_{test,1}^{\mathsf{Adv}}(\rho_{test,1} \otimes \frac{1}{D_2}\mathbb{I} \otimes \rho_{in}))) > \delta$$

By the soundness of $\pi_{test,1}$ there exists efficiently-computable simulator $\mathsf{Sim}_1^{\mathsf{Adv}}$ such that

$$\Pi_{\mathsf{pass}}(\pi_{comp,1}^{\mathsf{Adv}}(\Phi_1 \otimes \frac{1}{D_2}\mathbb{I} \otimes \rho_{in})) \approx_{\epsilon_1}^{ind} \Pi_{\mathsf{pass}}(\mathsf{Sim}_1^{\mathsf{Adv}}(\mathcal{E}_1(\Phi_1) \otimes \frac{1}{D_2}\mathbb{I} \otimes \rho_{in}))) \tag{21}$$

Now we move to analyze the second ROAV. First from (20) considering roundtype $= 2$ we get

$$\mathrm{tr}(\Pi_{\mathsf{pass}}(\pi_{test,2}^{\mathsf{Adv}}(\pi_{comp,1}^{\mathsf{Adv}}(\frac{1}{D_1}\mathbb{I} \otimes \rho_{test,2} \otimes \rho_{in})))) > 1 - 2(1 - \delta') \tag{22}$$

Notice the server-side of $\Phi_1$ in (21) is $\frac{1}{D_1}\mathbb{I}$, we can re-write (22) as

$$\mathrm{tr}(\Pi_{\mathsf{pass}}(\pi_{test,2}^{\mathsf{Adv}}(\pi_{comp,1}^{\mathsf{Adv}}(\Phi_1 \otimes \rho_{test,2} \otimes \rho_{in})))) > 1 - 2(1 - \delta') \tag{23}$$

Combining it with (21) we get

$$\mathrm{tr}(\Pi_{\mathsf{pass}}(\pi_{test,2}^{\mathsf{Adv}}(\mathsf{Sim}_1^{\mathsf{Adv}}(\mathcal{E}_1(\Phi_1) \otimes \rho_{test,2} \otimes \rho_{in})))) > 1 - 2(1 - \delta') - \epsilon_1 > \delta \tag{24}$$

Applying the soundness property of $(\rho_{test,2}, \pi_{test,2})$ we know there exists an efficiently computable server-side simulator $\mathsf{Sim}^{\mathsf{Adv}}$ such that

$$\Pi_{\mathsf{pass}}(\pi_{\mathsf{comp},2}^{\mathsf{Adv}}(\mathsf{Sim}_1^{\mathsf{Adv}}(\mathcal{E}_1(\Phi_1) \otimes \Phi_2 \otimes \rho_{in}))) \approx_{\epsilon_2}^{ind} \Pi_{\mathsf{pass}}(\mathsf{Sim}^{\mathsf{Adv}}(\mathcal{E}_1(\Phi_1) \otimes \mathcal{E}_2(\Phi_2) \otimes \rho_{in})) \tag{25}$$

Combining (21) and (25) completes the proof. □

### 4.1.4 Comparison to the non-local games setting

Recall that in the usual setting of self-testing protocols, the client sends questions to two spatially-separated quantum servers. How is this related to our notions? In our notion there is no explicit appearance of two different servers; however, we could think about what the server is able to do if we focus on a specific server in the multi-server setting: the state that it holds is determined by the questions to and answers from the other server, which is unknown to it; to pass the client's checking, it has to apply the specific operation, regardless what the underlying state is. Indeed, one reason that self-testing in the multi-server setting is powerful is it allows us to design protocols that behave as follows:

1. The client chooses to play either Game 1 or Game 2 with the two servers; the distribution of questions seems the same in the view of a specific server.

Game 1 is to control the server's operation and Game 2 is to perform some nontrivial cryptographic tasks. Then the soundness proof could go as follows:

1. The ability of passing Game 1 implies the servers' operations are close to some target operations.
2. Each of the servers is not aware of which game they are playing, and each of their operations only depends on the question it receives. Thus the operation closeness properties derived from Game 1 could be used to argue about behaviors of servers in Game 2.
3. Prove that servers with these behaviors could achieve the goal in Game 2.

Our notion shares the same intuition with the self-testing in the multi-server setting as described above. In our definition of ROAV, the $(\rho_{test}, \pi_{test})$ is used to test the server's behavior, and the soundness allows us to argue about the behavior of the server in $\pi_{comp}$.

## 4.2 Building RSPV from ROAV

In this subsection we argue that ROAV is potentially useful for building RSPV for state families that are not easy to construct directly. We give a protocol for building RSPV protocols from ROAV and more basic RSPV.

As a preparation we formulate a condition on the target state (Equation (3)).

**Definition 4.5.** Consider a target state (formulated in (3)):

$$\rho_{tar} = \sum_{i\in[D]} p_i \underbrace{|i\rangle\langle i|}_{\text{client}} \otimes \underbrace{|\varphi_i\rangle\langle\varphi_i|}_{\text{server}} \tag{26}$$

If $\{|\varphi_i\rangle\}_{i\in[D]}$ is an orthogonal normal basis, we say (26) is a target state with respect to an orthogonal normal basis.

**Protocol 2.** *Suppose $(\rho_{test}, \pi_{test}, \pi_{comp})$ is an ROAV for target operator $\mathcal{E}$. $p$ is a constant in $(0,1)$. $\rho_0$ is a target state as formulated in (26). Suppose $\pi_0$ is an RSPV for the target state*

$$p \underbrace{|\mathsf{test}\rangle}_{roundtype} \langle\mathsf{test}| \otimes \rho_{test} + (1-p)\,|\mathsf{comp}\rangle\langle\mathsf{comp}| \otimes \rho_0 \tag{27}$$

*where the roundtype register is on the client side, and the server-side of $\rho_{test}$ and $\rho_0$ are of the same dimension.*

1. *Run protocol $\pi_0$.*
2. *Depending on the value of roundtype:*
   - *If roundtype* $= \textsf{test}$*, run* $\pi_{test}$*.*
   - *If roundtype* $= \textsf{comp}$*, run* $\pi_{comp}$ *and keeps the output.*

**Theorem 4.2.** *Suppose $(\rho_{test}, \pi_{test}, \pi_{comp})$ is an ROAV for $\mathcal{E}$ with soundness error $\delta$ and approximation error $\epsilon$. $\pi_0$ is an RSPV for* (27) *with soundness error $\delta$ and approximation error $\epsilon_0$. Then Protocol 2 is an RSPV for target state $\mathcal{E}(\rho_0)$ with soundness error $\delta' = 1 - p(1-\delta) + \epsilon_0$ and approximation error $\epsilon' = 4p + \epsilon + \epsilon_0$.*

Thus to use this protocol we need to make $p$ small to keep the approximation error small, which leads to an RSPV protocol with large soundness error. But this could be solved by taking Protocol 2 to the cut-and-choose amplification protocol in Section 3.2.2.

The following fact is useful for proving Theorem 4.2.

**Fact 3.** *Suppose $|\Phi\rangle = \sum_{i\in[D]} \frac{1}{\sqrt{D}} |i\rangle \otimes |i\rangle$, $(|\varphi_1\rangle, |\varphi_2\rangle, \cdots |\varphi_D\rangle)$ is an orthogonal normal basis, $U \in \mathbb{C}^{D\times D}$ is defined as $U|i\rangle = |\varphi_i\rangle$, then*

$$(U^\dagger \otimes I)|\Phi\rangle = (I \otimes U)|\Phi\rangle$$

A corollary is a state in the form of (26) could be prepared by operating on the client side of $|\Phi\rangle$.

*Proof for Theorem 4.2.* Suppose the adversary is $\mathsf{Adv}$, the input state is $\rho_{in}$ and Protocol 2 passes with probability $> \delta'$. More formally, denoting the step 2 in Protocol 2 as $\pi_{\text{step2}}$, there is

$$\operatorname{tr}(\Pi_{\mathsf{pass}}((\pi_{\text{step2}} \circ \pi_0)^{\mathsf{Adv}}(\rho_{in}))) > \delta'. \tag{28}$$

First this implies the $\pi_0$ step passes with probability $> \delta' > \delta$. By the soundness of $\pi_0$ there exists an efficiently computable server-side simulator $\mathsf{Sim}_0^{\mathsf{Adv}}$ such that

$$\Pi_{\mathsf{pass}}(\pi_0^{\mathsf{Adv}}(\rho_{in})) \approx^{ind}_{\epsilon_0} \Pi_{\mathsf{pass}}(\mathsf{Sim}_0^{\mathsf{Adv}}((27) \otimes \rho_{in})) \tag{29}$$

This together with (28) implies

$$\operatorname{tr}(\Pi_{\mathsf{pass}}(\pi_{\text{step2}}^{\mathsf{Adv}}(\mathsf{Sim}_0^{\mathsf{Adv}}((27) \otimes \rho_{in})))) > \delta' - \epsilon_0. \tag{30}$$

which further implies

$$\operatorname{tr}(\Pi_{\mathsf{pass}}(|\mathsf{test}\rangle\langle\mathsf{test}| \otimes (\pi_{test}^{\mathsf{Adv}}(\mathsf{Sim}_0^{\mathsf{Adv}}(\rho_{test} \otimes \rho_{in}))))) > \delta \tag{31}$$

From (31), by the ROAV soundness there exists an efficiently computable server-side simulator $\mathsf{Sim}^{\mathsf{Adv}}$ such that

$$\Pi_{\mathsf{pass}}(\pi_{\mathsf{comp}}^{\mathsf{Adv}}(\mathsf{Sim}_0^{\mathsf{Adv}}(\Phi \otimes \rho_{in}))) \approx^{ind}_{\epsilon} \Pi_{\mathsf{pass}}(\mathsf{Sim}^{\mathsf{Adv}}(\mathcal{E}(\Phi) \otimes \rho_{in}))$$

Applying Fact 3 we can measure the client-side of $\Phi$ to collapse it to $\rho_0$:

$$\Pi_{\mathsf{pass}}(\pi_{\mathsf{comp}}^{\mathsf{Adv}}(\mathsf{Sim}_0^{\mathsf{Adv}}(\rho_0 \otimes \rho_{in}))) \approx^{ind}_{\epsilon} \Pi_{\mathsf{pass}}(\mathsf{Sim}^{\mathsf{Adv}}(\mathcal{E}(\rho_0) \otimes \rho_{in})) \tag{32}$$

Notice that $\mathsf{Sim}_0^{\mathsf{Adv}}(\rho_0 \otimes \rho_{in}) \approx_{2p} \mathsf{Sim}_0^{\mathsf{Adv}}((27) \otimes \rho_{in})$, this together with (29)(32) implies

$$\Pi_{\mathsf{pass}}((\pi_{\mathsf{comp}} \circ \pi_0)^{\mathsf{Adv}}(\rho_{in})) \approx^{ind}_{2p+\epsilon_0+\epsilon} \Pi_{\mathsf{pass}}(\mathsf{Sim}^{\mathsf{Adv}}(\mathcal{E}(\rho_0) \otimes \rho_{in}))$$

Noticing that $\pi_{\mathsf{comp}}(\cdot) \approx_{2p} \pi_{\text{step2}}(\cdot)$ completes the proof. □

## 4.3 Testing Ground State Energy by ROAV

In this subsection we give a Hamiltonian ground energy testing protocol based on RSPV and ROAV.

### 4.3.1 Overview of the protocol

As a review, in existing Hamiltonian ground energy testing protocols like [9, 11], the high level structure of protocols is typically as follows:

Input: a Hamiltonian $H = \sum_i \gamma_i H_i$ where $H_i$ is simple.

The honest server gets a witness state $\rho$.

1. Repeat (sequentially or in parallel) the following for polynomial number of times:

   The client samples a random $H_i$ and uses some protocols to get the measurement results of operator $H_i$ on the server-side state. The server is not able to know which operator the client is measuring.

2. The client calculates the weighted average of the measurement results in the first step and decides whether it's a yes-instance or no-instance.

The first step seems to have a form of ROAV, in the sense that the protocols aim at certifying that the server has measured an operator obliviously. Although the witness state is not held by the client, this is still within reach of our definition in Section 4.1.2. A more subtle difference here is that an ROAV protocol is defined for a fixed family of operators, while in the protocol described above the target operators depends on the input Hamiltonian $H$. We expect that it's typically harder to construct ROAV compared to other primitives like RSPV, thus we want to find a way to reduce this task to simpler primitives.

In this subsection we show a protocol that reduces this problem to the following two protocols:

1. An ROAV for a simple, fixed operator family: tensor products of Bell basis measurements.

2. An RSPV for state families that depend on the input Hamiltonian (but still as simple as products of simple states).

The idea is to make use of teleportation-based computation [6].[3] As a simple example, we consider the single-qubit witness case below.

As the setup, assume the server holds a single-qubit witness state $\rho$, and in addition holds one of the four Bell state (see Definition 2.1). Index the qubit register for the witness with wire number $w = 1$, and index the qubit register for the Bell state with wire number $w = 2, 3$. Then the quantum teleportation says a Bell-basis measurement on qubit $1, 2$ results in a state of the following form on qubit 3:

$$\mathsf{X}^{a'}\mathsf{Z}^{b'}(\rho), a', b' \text{ depend on the Bell state choices and measurement outcomes}$$

An explicit expression for this process is as follows. Use $\mathcal{E}$ to denote the Bell-basis measurement, use $\mathsf{X}^a\mathsf{Z}^b\,|\varphi\rangle$ to denote different Bell basis states, there is

$$(\mathcal{E}\otimes \mathsf{I})(\rho\otimes \mathsf{X}^a\mathsf{Z}^b(\varphi)) = \sum_{c,d\in\{0,1\}^2}\frac{1}{4}\underbrace{|c,d\rangle}_{\text{measurement outcome}}\langle c,d|\otimes \mathsf{X}^{a+c}\mathsf{Z}^{b+d}(\rho)$$

[3] Several other existing works [11, 5] also use it for different purposes or in different settings.

Then the standard basis measurement outcome on these three qubits encodes the standard basis measurement outcome on $\rho$. Furthermore, to control the measurement operator applied on $\rho$, the client only needs to have control on the state in qubit number 3, as follows:

$$\text{For any gate } g \text{ on the 3rd qubit, } (\mathcal{E}\otimes \mathsf{I})(\rho\otimes g(\mathsf{X}^a\mathsf{Z}^b(\varphi))) = \sum_{c,d\in\{0,1\}^2}\frac{1}{4}\left|c,d\right\rangle\left\langle c,d\right|\otimes g(\mathsf{X}^{a+c}\mathsf{Z}^{b+d}(\rho)) \tag{33}$$

Especially, if $g = \mathsf{H}$, $g(\mathsf{X}^{a+c}\mathsf{Z}^{b+d}(\rho)) = \mathsf{X}^{b+d}\mathsf{Z}^{a+c}(g(\rho))$.

This relation allows us to reduce the task of measuring operator $H_i$ on $\rho$ to the standard basis measurement of (33). What we need for translating (33) to a protocol is an ROAV for $\mathcal{E}$ and an RSPV for the states that will be used (including the test state of ROAV, $g(\mathsf{X}^a\mathsf{Z}^b(\rho))$ for $g\in\{\mathsf{I},\mathsf{H}\}$). Below we formulate the protocol.

### 4.3.2 Protocol formulation

To formulate the protocol, we define several notations for preparation.

**Notation 4.1** (Notation preparation for Protocol 3)**.** Consider qubit registers indexed by $(1,i),(2,i),(3,i)$, $i\in[n]$. Define

$$\left|\varphi\right\rangle = \otimes_{i\in[n]}\left(\frac{1}{\sqrt{2}}\left(\left|\underbrace{0}_{(2,i)}\underbrace{0}_{(3,i)}\right\rangle + \left|\underbrace{1}_{(2,i)}\underbrace{1}_{(3,i)}\right\rangle\right)\right) \tag{34}$$

Define index set $I_2 = \{(2,i)|i\in[n]\}$, and define $I_3$ similarly. We say $\vec{a}\in\{0,1\}^n$ indexed by $I_2$, when its coordinates are denoted as as $a_{(2,i)}$ where $(2,i)\in I_2$. Define $\mathsf{X}^{\vec{a}}$ as the operation that applies $\mathsf{X}$ on qubit $(2,i)$ if $a_{(2,i)}=1$. Define notations $\mathsf{Z}^{\vec{b}}$ and $\mathsf{H}^{\vec{v}}$ similarly.

Define the following state in the form of (3), which is the family of all the possible four Bell states on wire $2,3$ for each $i$:

$$\left|\phi\right\rangle\left\langle\phi\right| = \frac{1}{2^{2n}}\sum_{\vec{a},\vec{b}\in\{0,1\}^n\text{ indexed by } I_2}\underbrace{\left|\vec{a},\vec{b}\right\rangle\left\langle\vec{a},\vec{b}\right|}_{\text{client-side}}\otimes\mathsf{X}^{\vec{a}}\mathsf{Z}^{\vec{b}}(\varphi)$$

Define operation

$$\mathcal{E} = \text{“For each } i\text{, measure } (1,i),(2,i) \text{ on the Bell basis and measure } (3,i) \text{ on the standard basis and report the result to the client.”}$$

Below we introduce more notations that deal with the Hamiltonian and its repetition.

**Notation 4.2** (More notation preparation for Protocol 3)**.** For an XZ-local-Hamiltonian as defined in (4)(2)

$$H = \sum_{j\in[m]}\gamma_j H_j, \tag{35}$$

denote $\mathsf{vecx}(H_j)$ as an $n$-dimension vector indexed by $I_3$ that:

- If the observable on the $i$-th qubit in $H_j$ is $\sigma_X$, the $i$-th coordinate of $\mathsf{vecx}(H_j)$ is 1;
- If the observable on the $i$-th qubit in $H_j$ is $\sigma_Z$ or $\mathsf{I}$, the $i$-th coordinate of $\mathsf{vecx}(H_j)$ is 0.

That is, $\mathsf{vecx}(H_j)$ indicates whether the corresponding observable in $H_j$ is the $\sigma_\mathsf{X}$ observable.

Similarly define $\mathsf{vecz}(H_j)$ as the indicator for whether the corresponding observable in $H_j$ is the $\sigma_\mathsf{Z}$ observable.

An example is as follows: $\mathsf{H}^{\mathsf{vecx}(H_i)}$ flips all the $\sigma_X$ operations in $H_i$ to $\sigma_Z$ operations and keeps the others unchanged. Then define state

$$\rho_{comp} = \frac{1}{m}\sum_{j\in[m]} \underbrace{|j\rangle\langle j|}_{\text{client-side}} \otimes \mathsf{H}^{\mathsf{vecx}(H_j)}(\phi) \tag{36}$$

That is, the client randomly samples an $H_j$ from (35) and flips the $\sigma_\mathsf{X}$ operators to $\sigma_\mathsf{Z}$ operators.

Now consider the $K$-fold tensor product for the notations above and in Notation 4.1. The qubit registers are indexed by $(w,i,k)$, $w\in\{1,2,3\}$, $i\in[n]$, $k\in[K]$. Then $|\varphi\rangle^{\otimes K}$ is defined as the $k$-fold tensor product of $|\varphi\rangle$ arranged on registers $(2,i,k),(3,i,k)$. Similarly $|\phi\rangle^{\otimes K}$ is defined as the $k$-fold tensor product of $|\phi\rangle$, where the client-side registers are denoted by $\vec{a}_k,\vec{b}_k$, $k\in[K]$. $\mathcal{E}^{\otimes K}$ is similarly defined as applying $\mathcal{E}$ for each $k\in[K]$. Then similarly $\rho_{comp}^{\otimes K}$ is defined as

$$\sum_{\vec{j}=(j_1,j_2\cdots j_K)\in[m]^K} |\vec{j}\rangle\langle\vec{j}| \otimes \mathsf{H}^{\mathsf{vecx}(H_{j_1})}(\phi)\otimes \mathsf{H}^{\mathsf{vecx}(H_{j_2})}(\phi)\otimes\cdots\otimes \mathsf{H}^{\mathsf{vecx}(H_{j_K})}(\phi)$$

Then the honest behavior that we want to design a protocol for could be described as

$$\mathcal{E}^{\otimes K}(\rho^{\otimes K}\otimes\rho_{comp}^{\otimes K}) \tag{37}$$

where $\rho$ is the ground state of the Hamiltonian.

Then we define a series of notations for arguing about the energy corresponding to (37). For each $k\in[K]$, introduce variable $\vec{c}_k\in\{0,1\}^n,\vec{d}_k\in\{0,1\}^n,\vec{e}_k\in\{0,1\}^n$ (which are $3K$ $n$-dimensional vectors), and they correspond to the measurement outcome of (37) on qubits in the $k$-th fold indexed by $I_1,I_2,I_3$. Then define

$$\mathrm{valtemp}^H(\vec{a}_k,\vec{b}_k,\vec{c}_k,\vec{d}_k,\vec{e}_k,j_k) = ((\vec{a}_k+\vec{c}_k+\vec{e}_k)\cdot\mathsf{vecz}(H_{j_k}) + (\vec{b}_k+\vec{d}_k+\vec{e}_k)\cdot\mathsf{vecx}(H_{j_k})) \mod 2$$

$$\mathrm{val}^H(\vec{a}_k,\vec{b}_k,\vec{c}_k,\vec{d}_k,\vec{e}_k,j_k) = m\cdot\gamma_{j_k}(-1)^{\mathrm{valtemp}^H(\vec{a}_k,\vec{b}_k,\vec{c}_k,\vec{d}_k,\vec{e}_k,j_k)}$$

Use $T_k$ to denote the tuple $(\vec{a}_k,\vec{b}_k,\vec{c}_k,\vec{d}_k,\vec{e}_k,j_k)$ and use $T$ to denote the tuple $(T_k)_{k\in[K]}$. Define

$$\mathrm{val}^H(T) = \frac{1}{K}\sum_{k\in[K]}\mathrm{val}^H(T_k) \tag{38}$$

**Protocol 3.** *Input: an XZ 5-local Hamiltonian $H=\sum_{j\in[m]}\gamma_j H_j$, $a,b$, $b-a\geq 1/\mathsf{poly}(n)$, as Definition 2.3.*

*Take $K=100\kappa^2\frac{1}{(b-a)^2}$. Consider qubit registers indexed by $(w,i,k)$, $w\in\{1,2,3\},i\in[n],k\in[K]$. Use notations in Notation 4.1, 4.2. Suppose $(\rho_{test},\pi_{test},\pi_{comp})$ is an ROAV for $\mathcal{E}^{\otimes K}$. Suppose $\pi_0$ as an RSPV for the following target state:*

$$\frac{1}{2}\underbrace{|\mathsf{operatortest}\rangle}_{roundtype}\langle\mathsf{operatortest}|\otimes\rho_{test} + \frac{1}{2}|\mathsf{energytest}\rangle\langle\mathsf{energytest}|\otimes(\rho_{comp})^{\otimes K} \tag{39}$$

*Note the roundtype information is kept on the client side and hidden from the server.*

1. *Execute protocol $\pi_0$.*

2. *Depending on the value of roundtype:*

   - *If roundtype* $=$ operatortest, *the client executes $\pi_{test}$ with the server.*
   - *If roundtype* $=$ energytest, *the client executes $\pi_{comp}$ with the server. Suppose the set of client-side information is denoted by $T$ as in Notation 4.2. Accept if val$(T) \leq \frac{a+b}{2}$ and reject otherwise.*

**Theorem 4.3.** *Suppose $(\rho_{test}, \pi_{test}, \pi_{comp})$ is complete, $\pi_0$ is complete, then Protocol 3 is complete.*

*Proof.* From the completeness we know the first step of Protocol 3 succeeds with $1 - \mathsf{negl}(\kappa)$ probability, the operatortest succeeds with $1 - \mathsf{negl}(\kappa)$ probability, and the energy test implements (37) up to only a negligible error. For a yes-instance, for each $k \in [K]$, by the promise there is $\mathbb{E}[\mathrm{val}^H(T_k)] \leq a$. Thus for $K = 100\kappa^2 \cdot \frac{1}{(b-a)^2}$ by Chernoff's bound there is $\Pr[\mathrm{val}^H(T) \leq \frac{a+b}{2}] \leq 2^{-\kappa}$. □

**Theorem 4.4.** *Suppose $(\rho_{\mathsf{test}}, \pi_{\mathsf{test}}, \pi_{\mathsf{comp}})$ is an ROAV with soundness error $\delta$ and approximation error $\epsilon$. $\pi_0$ is an RSPV with soundness error $\delta$ and approximation error $\epsilon_0$. Then Protocol 3 has soundness error $\delta' = \min\{1 - \frac{1}{2}(1-\delta) + 2\epsilon_0, \epsilon + \epsilon_0 + \frac{1}{2} + \mathsf{negl}(\kappa)\}$.*

By substituting suitable parameters it's possible to make the soundness smaller than the completeness.

*Proof.* Suppose $H$ is a no-instance, the adversary is $\mathsf{Adv}$ and the protocol passes with probability $> \delta'$. That is, [4]

$$\mathrm{tr}(\Pi_{\mathsf{pass}}(\pi_{\mathrm{step2}} \circ \pi_0)^{\mathsf{Adv}}) > \delta' \tag{40}$$

This implies $\mathrm{tr}(\Pi_{\mathsf{pass}}(\pi_0^{\mathsf{Adv}})) > \delta$. By the soundness property of $\pi_0$ there exists an efficiently computable server-side simulator $\mathsf{Sim}_0^{\mathsf{Adv}}$ such that

$$\Pi_{\mathsf{pass}}(\pi_0^{\mathsf{Adv}}) \approx_{\epsilon_0}^{ind} \Pi_{\mathsf{pass}}(\mathsf{Sim}_0^{\mathsf{Adv}}(\text{equation (39)})) \tag{41}$$

Further note (40) implies $\mathrm{tr}(\Pi_{\mathsf{pass}}\Pi_{\mathsf{operatortest}}(\pi_{\mathrm{step2}} \circ \pi_0)^{\mathsf{Adv}}) > \delta'$. Combining it with (41) we get

$$\mathrm{tr}(\Pi_{\mathsf{pass}}\pi_{\mathsf{test}}^{\mathsf{Adv}\circ\mathsf{Sim}_0^{\mathsf{Adv}}}(\rho_{\mathsf{test}})) > 1 - 2(1 - \delta' + \epsilon_0) > \delta \tag{42}$$

By the soundness of ROAV this implies there exists an efficiently computable server-side simulator $\mathsf{Sim}^{\mathsf{Adv}}$, a server-side state $\rho$ (corresponding to the output of $\mathsf{Sim}_{in}^{\mathsf{Adv}}$ in the soundness of ROAV) such that

$$\Pi_{\mathsf{pass}}(\pi_{\mathsf{comp}}^{\mathsf{Adv}\circ\mathsf{Sim}_0^{\mathsf{Adv}}}(\Phi)) \approx_{\epsilon}^{ind} \Pi_{\mathsf{pass}}(\mathsf{Sim}^{\mathsf{Adv}}(\mathcal{E}(\rho \otimes \Phi))) \tag{43}$$

Now we consider a distinguisher that first measures the client side of $\Phi$ and collapse both sides of (43) to connect (43) with $\rho_{comp}$. This is done as follows:

1. First sample the client-side register corresponding to $\vec{j}$ in $\rho_{comp}$;

2. Then by Fact 3 there is a client-side measurement that collapses the server-side to $\mathsf{H}^{\mathsf{vecx}(H_j)}(\phi)$ (see (36)).

[4] We omit the initial states since it's not important here and could be $|0\rangle$.

This implies

$$\Pi_{\mathsf{pass}}(\pi_{\mathsf{comp}}^{\mathsf{Adv}\circ\mathsf{Sim}_0^{\mathsf{Adv}}}(\rho_{\mathsf{comp}}^{\otimes K})) \approx_{\epsilon}^{ind} \Pi_{\mathsf{pass}}(\underbrace{|\mathsf{energytest}\rangle\langle\mathsf{energytest}|}_{\text{roundtype}} \otimes(\mathsf{Sim}^{\mathsf{Adv}}(\mathcal{E}(\rho\otimes\rho_{\mathsf{comp}}^{\otimes K})))) \qquad (44)$$

Porjecting (41) onto roundtype $=$ $\mathsf{comp}$ we get

$$\mathrm{tr}(\Pi_{\mathsf{pass}}(\pi_{\mathrm{step2}}\circ\pi_0)^{\mathsf{Adv}})) \leq 1-\frac{1}{2}(1-\mathrm{tr}(\Pi_{\mathsf{pass}}(|\mathsf{energytest}\rangle\langle\mathsf{energytest}|\otimes(\pi_{\mathsf{comp}}^{\mathsf{Adv}}(\mathsf{Sim}_0^{\mathsf{Adv}}(\rho_{\mathsf{comp}}^{\otimes K}))))))+\epsilon_0$$

Combining it with the left hand side of (44) we get

$$\delta' \leq \frac{1}{2}\,\mathrm{tr}(\Pi_{\mathsf{pass}}(|\mathsf{energytest}\rangle\langle\mathsf{energytest}| \otimes (\mathsf{Sim}^{\mathsf{Adv}}(\mathcal{E}(\rho\otimes\rho_{\mathsf{comp}}^{\otimes K}))))) + \epsilon_0 + \epsilon + \frac{1}{2} \qquad (45)$$

Now we analyze the energy test passing probability in (45). By the definition of $\mathcal{E}$ and $\rho_{\mathsf{comp}}^{\otimes K}$, $\mathcal{E}(\rho\otimes\rho_{\mathsf{comp}}^{\otimes K})$ is applying the energy test described in the beginning of Section 4.3.1, and see whether $\mathrm{val}^H(T) < \frac{a+b}{2}$ holds (see (38) for the definition of $\mathrm{val}(T)$). By the fact that the ground energy of $H$ is $\geq b$, we know for each $k\in[K]$, conditioned on any possible outcome of $T_1, \cdots T_{k-1}$, there is $\mathbb{E}[\mathrm{val}^H(T_k)] \geq b$; then we could apply the Chernoff's bound and get

$$\mathrm{tr}(\Pi_{\mathrm{val}^H(T)<\frac{a+b}{2}}(\mathcal{E}(\rho\otimes\rho_{\mathsf{comp}}^{\otimes K}))) < 2^{-\kappa}$$

Substituting it to (45) completes the proof. □

# References

[1] Alexander Poremba Alexandru Gheorghiu, Tony Merger. Quantum cryptography with classical communication: parallel remote state preparation for copy-protection, verification, and more. 2022.

[2] James Bartusek. Secure quantum computation with classical communication. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography*, pages 1–30, Cham, 2021. Springer International Publishing.

[3] Charles H. Bennett, David P. DiVincenzo, Peter W. Shor, John A. Smolin, Barbara M. Terhal, and William K. Wootters. Remote state preparation. *Phys. Rev. Lett.*, 87:077902, Jul 2001.

[4] Zvika Brakerski, Alexandru Gheorghiu, Gregory D. Kahanamoku-Meyer, Eitan Porat, and Thomas Vidick. Simple tests of quantumness also certify qubits. 2023.

[5] Zvika Brakerski and Henry Yuen. Quantum garbled circuits. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 804–817, New York, NY, USA, 2022. Association for Computing Machinery.

[6] Nai-Hui Chia, Chia-Hung Chien, Wei-Ho Chung, and Sy-Yen Kuo. Quantum blind computation with teleportation-based computation. In *2012 Ninth International Conference on Information Technology - New Generations*, pages 769–774, 2012.

[7] Alexandru Cojocaru, Léo Colisson, Elham Kashefi, and Petros Wallden. Qfactory: Classically-instructed remote secret qubits preparation. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I*, volume 11921 of *Lecture Notes in Computer Science*, pages 615–645. Springer, 2019.

[8] Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum homomorphic encryption for polynomial-sized circuits. *IACR Cryptol. ePrint Arch.*, 2016:559, 2016.

[9] Joseph F. Fitzsimons, Michal Hajdušek, and Tomoyuki Morimae. Post hoc verification of quantum computation. *Phys. Rev. Lett.*, 120:040501, Jan 2018.

[10] Alexandru Gheorghiu and Thomas Vidick. Computationally-secure and composable remote state preparation. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 1024–1033. IEEE Computer Society, 2019.

[11] Alex B. Grilo. A Simple Protocol for Verifiable Delegation of Quantum Computation in One Round. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 28:1–28:13, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[12] Qi Zhao Honghao Fu, Daochen Wang. Computational self-testing of multi-qubit states and measurements. 2022.

[13] Zhengfeng Ji. Classical verification of quantum proofs. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 885–898, New York, NY, USA, 2016. Association for Computing Machinery.

[14] Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. Mip* = re. *Commun. ACM*, 64(11):131–138, oct 2021.

[15] Gregory D. Kahanamoku-Meyer, Soonwon Choi, Umesh V. Vazirani, and Norman Y. Yao. Classically verifiable quantum advantage from a computational Bell test. *Nature Phys.*, 18(8):918–924, 2022.

[16] Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Lisa Yang. Quantum advantage from any non-local game. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 1617–1628, New York, NY, USA, 2023. Association for Computing Machinery.

[17] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition.* Chapman & Hall/CRC, 2nd edition, 2014.

[18] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 332–338. IEEE Computer Society, 2018.

[19] Urmila Mahadev. Classical verification of quantum computations. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 259–267. IEEE Computer Society, 2018.

[20] Dominic Mayers and Andrew Yao. Self testing quantum apparatus. *Quantum Info. Comput.*, 4(4):273–286, jul 2004.

[21] Tony Metger and Thomas Vidick. Self-testing of a single quantum device under computational assumptions. In *ITCS*, 2021.

[22] Akihiro Mizutani, Yuki Takeuchi, Ryo Hiromasa, Yusuke Aikawa, and Seiichiro Tani. Computational self-testing for entangled magic states. *Phys. Rev. A*, 106(1):L010601, 2022.

[23] Anand Natarajan and Tina Zhang. Bounding the quantum value of compiled nonlocal games: from chsh to bqp verification. 2023.

[24] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, New York, NY, USA, 10th edition, 2011.

[25] Arun Kumar Pati. Minimum cbits required to transmit a qubit. *Phys. Rev. A*, 63:014320, 2001.

[26] Sandu Popescu and Daniel Rohrlich. Which states violate bell's inequality maximally? *Physics Letters A*, 169(6):411–414, 1992.

[27] Ben Reichardt, Falk Unger, and Umesh V. Vazirani. Classical command of quantum systems. *Nature*, 496:456–460, 2013.

[28] Gregory Rosenthal and Henry S. Yuen. Interactive proofs for synthesizing quantum states and unitaries. In *Information Technology Convergence and Services*, 2021.

[29] Stephen J. Summers and R. Werner. Maximal Violation of Bell's Inequalities Is Generic in Quantum Field Theory. *Commun. Math. Phys.*, 110:247–259, 1987.

[30] J. Zhang. Classical verification of quantum computations in linear time. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 46–57, Los Alamitos, CA, USA, nov 2022. IEEE Computer Society.

[31] Jiayu Zhang. *Succinct Blind Quantum Computation Using a Random Oracle*, page 1370–1383. Association for Computing Machinery, New York, NY, USA, 2021.