Formulations and Constructions of Remote State Preparation with Verifiability, with Applications

Jiayu Zhang ✉
Zhongguancun Laboratory, Beijing, China

Abstract

Remote state preparation with verifiability (RSPV) is an important quantum cryptographic primitive [20, 41]. In this primitive, a client would like to prepare a quantum state (sampled or chosen 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. In this work we make several contributions on its formulations, constructions and applications. In more detail:

- We first work on the definitions and abstract properties of the RSPV problem. We select and compare different variants of definitions [8, 20, 41, 19], and study their basic properties (like composability and amplification).
- We also study a closely related question of how to certify the server's operations (instead of solely the states). We introduce a new notion named *remote operator application with verifiability* (ROAV). We compare this notion with related existing definitions [38, 28, 24, 29, 31], study its abstract properties and leave its concrete constructions for further works.
- Building on the abstract properties and existing results [11], we construct a series of new RSPV protocols. Our constructions not only simplify existing results [20] but also cover new state families, for example, states in the form of $\frac{1}{\sqrt{2}}(|0\rangle |x_0\rangle + |1\rangle |x_1\rangle)$. All these constructions rely only on the existence of weak NTCF [12, 5], without additional requirements like the adaptive hardcore bit property [10, 5].
- As a further application, we show that the classical verification of quantum computations (CVQC) problem [2, 27] could be constructed from assumptions on group actions [4]. This is achieved by combining our results on RSPV with group-action-based instantiation of weak NTCF [5], and then with the quantum-gadget-assisted quantum verification protocol [17].

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases Quantum Cryptography, Remote State Preparation, Self-testing, Verification of Quantum Computations

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.96

Related Version *Full Version*: `https://arxiv.org/abs/2310.05246`

Funding *Jiayu Zhang*: 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.

Acknowledgements I would like to thank Thomas Vidick, Zhengfeng Ji, Anne Broadbent and Qipeng Liu for discussions.

1 Introduction

1.1 Background

Development of quantum computers [6, 42, 9] leads to demands of various quantum cryptographic protocols. In a typical setting, there are a client and a remote quantum server. The client would like to achieve some quantum tasks, but it does not trust the server; thus the client would like to make use of cryptography to achieve its goal. Famous examples include quantum computation verification [27, 41], multiparty quantum computations [7], etc. In this work, we are interested in a basic and very important primitive called *remote state preparation* (RSP) [8], which we introduce below.

1.1.1 Remote state preparation: an overview

In the RSP problem, ideally, the client would like to prepare a quantum state (sampled or chosen 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 achieve this task using cheaper resources (for example, only classical communication), possibly under computational assumptions?

Studies of RSP have a long history [32, 8]. One setting of RSP is the fully honest setting [8]. In this work, we are interested in the cryptographic setting where the server could be malicious. Then a formulation of RSP should at least have a correctness requirement and a security requirement.

The natural correctness requirement for RSP says that when the server is honest, the server gets the state while the client gets the state description. For security, there are different security notions, including blindness (or privacy, secrecy) and verifiability (or soundness) [15, 20, 40]. In this paper we focus on RSP with verifiability (RSPV). In RSPV, intuitively, the client is able to verify that in the case of passing (or called acceptance, non-aborting) the server (approximately) really gets the state, as if it is sent through a quantum channel. A malicious server who attempts to get other information 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. For example, [20] first constructs a classical channel cryptography-based RSPV and uses it to achieve classical verification of quantum computations; [19] explores more applications of RSPV; [41] takes the RSPV approach to achieve classical verification of quantum computations with linear total time complexity. Many quantum cryptographic protocols rely on the quantum channel and quantum communication, and an RSPV protocol could serve as a compiler: it allows us to replace these quantum communication steps by other cheaper resources, like the classical communication.

On the one hand, there have been many important and impressive results in this direction; on the other hand, there are also various limitations or subtleties in existing works, including formulations and constructions. Below we discuss existing works in more detail and motivate our results.

1.2 Existing Works and Motivating Questions

1.2.1 Formulations and abstract properties of RSPV

We first note that there are many variants of definitions for RSPV. For example, there are two subtlely different types of security notions, the *rigidity-based* (or isometry-based) soundness [15, 19] and *simulation-based* soundness [8, 20, 41]. Existing works do not seem to care about the differences; we note that these differences could have impact on the abstract properties of the definitions and could affect their well-behaveness. For example, we would like RSPV to have sequential composability between independent instances: if the client and the server execute an RSPV protocol for a state family $\mathcal{F}_1$, and then execute an RSPV protocol for a state family $\mathcal{F}_2$, we would like the overall protocol to be automatically an RSPV for $\mathcal{F}_1 \otimes \mathcal{F}_2$. If such a sequential composability property holds, protocols for tensor products of states could be reduced to protocols for each simple state family.

In this background, we argue that:

It's helpful to compare variants of definitions and formalize basic abstract properties.

We would like to have a more well-behaved framework for RSPV, which will lay a solid foundation for concrete constructions.

1.2.2 Definitions of the primitive for certifying the server's operations

RSPV talks about the certification of server-side *states*. A closely related question is: how could the client certify the server-side *operations*?

To address this problem, existing works raise the notion of self-testing [38, 33, 28]. One famous scenario of self-testing is in non-local games [24, 35]. In this scenario, the verifier sends questions to two spatially-separated but entangled quantum provers (or called servers). The verifier's questions and passing conditions are specially designed so that the provers have to perform specific operations to pass. This provides a way to constrain the provers' operations through only classical interactions and spatial separation, which has become a fundamental technique in the study of non-local games.

Recently a series of works [29, 18, 11, 31] study the single-server cryptographic analog of the non-local game self-testing. [29] studies the cryptographic analog of the CHSH game, where the server needs to prepare the Bell states and perform measurements on two of its registers. [30, 18] further extend it to three-qubit and N-qubit; [25, 31] make use of QFHE [26] to formulate and address the problem, where the FHE-encrypted part takes the role of one prover and the unencrypted part takes the role of the other prover.

What's common in these existing works is that they are defining the single-server cryptographic analog of the non-local game self-testing to be a protocol where the single server is playing the roles of *both* of the two provers. In this work we are interested in another viewpoint, which is:

How could we define a single-server cryptographic analog of the non-local game self-testing, where the single server is playing the role of one *of the two provers?*

1.2.3 Existing RSPV constructions

Maybe the most natural question in the direction of RSPV is:

For what state families could we achieve RSPV?

Let's quickly review what state families have been achieved in existing works. [20] achieves RSPV for $\{|+_\theta\rangle := \frac{1}{\sqrt{2}}(|0\rangle + e^{\mathrm{i}\theta\pi/4}|1\rangle), \theta \in \{0, 1, 2 \cdots 7\}\}$; independently [15] gives a candidate RSPV construction for this state family and conjectures its security. [20] achieves RSPV for tensor products of BB84 states: $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}^{\otimes n}$. [41] achieves RSPV for tensor products of $|+_\theta\rangle$ states. [11] achieves RSPV for BB84 states.

For a broader viewpoint let's also review some famous self-testing protocols. [29] achieves cryptographic self-testing for Bell states and the corresponding X/Z measurements. [30] achieves self-testing for a 3-qubit magic state and the corresponding measurements. [18] achieves self-testing for multiple Bell pairs and measurements. [31] achieves self-testing for "all-X" and "all-Z" operations.

As a summary, one significant limitations of existing works is that they could only handle simple tensor product states and operators. We consider this situation very undesirable since ideally we want a protocol that is sufficiently powerful to cover all the computationally-efficient state families. The lack of concrete protocols also restricts the applications of RSPV as a protocol compiler: suppose that we have a quantum cryptographic protocol that starts with sending states in (for example) $\{\frac{1}{\sqrt{2}}(|0\rangle |x_0\rangle + |1\rangle |x_1\rangle), x_0, x_1 \in \{0,1\}^n\}$, it's not clear how to use existing RSPV protocols to compile this quantum communication step to the classical communication. (Note that although there are mature techniques for creating this type of states with some other security notions [26, 10, 27], it's not known how to construct RSPV for it.)

Besides the existency problem, it's also desirable if we could weaken the assumptions needed or simplify existing results.

1.2.4 Classical verification of quantum computations and its assumption

We will apply our results on RSPV to the classical verification of quantum computations (CVQC) problem. In this problem a classical client would like to delegate the evaluation of a BQP circuit C to a quantum server, but it does not trust the server; thus it wants to make use of a protocol to verify the results. The first and perhaps the most famous construction is given by Mahadev [27]. Building on Mahadev's work and related results [26, 10, 27], a series of new CVQC protocols are constructed [3, 20, 14, 41, 31, 5].

However, one undesirable situation is that all the existing constructions are either based on LWE [34], or based on the random oracle heuristic, or based on some new conjecture on the reduction between security notions. In more detail:

- Most existing works [27, 14, 20] make use of NTCF with the "adaptive hardcore bit property" (or some stronger variants). The only known instantiation of this primitive from standard assumption is based on LWE.
- [5] gives an instantiation of NTCF with a weaker variant of the adaptive hardcore bit property using cryptographic group actions. However, to achieve the adaptive hardcore bit property it relies on an unproven conjecture on the reduction between these two notions. Alternatively one could make use of the random oracle to construct CVQC without the adaptive hardcore bit property [12, 41] but the instantiation of the random oracle is heuristic.
- If we only make use of NTCF without the adaptive hardcore bit property, we could do RSPV for BB84 states [11], but BB84 states are not known to be sufficient for constructing CVQC.
- [31] makes use of a quantum FHE-based approach to construct CVQC. However instantiations of quantum FHE rely on (variants of) NTCF and the classical FHE [26, 22], which still rely on the LWE assumption.

As mentioned above, a type of cryptographic assumption that is different from LWE is the cryptographic group actions, for example, supersingular isogeny [4, 13]. In this background, we ask:

Could we construct CVQC from cryptographic group actions?

1.3 Our Contributions

In this work we address or make progress to the questions above. Below we give an overview of our contributions; we refer to the full version for details.

1.3.1 Definitions and Abstract Properties of RSPV

We first work on the definitions and abstract properties of RSPV.

1. We first formulate and review the hierarchy of notions for formalizing RSPV. This includes the notions of registers, cq-states, protocols, paradigms of security definitions, etc.
2. We then formalize the notion of RSPV. We formulate the soundness as a simulation-based definition and compare this definition with several other variants (like the rigidity-based soundness, which is also popular).
3. We then study the abstract properties of RSPV including the sequential composability and amplification. This allows us to reduce the constructions of RSPV to primitives that are relatively easier to construct.

In summary, we clarify subtleties and build a well-behaved framework for RSPV, which enables us to build larger protocols from smaller or easier-to-construct components. In later part of this work we will build concrete RSPV constructions under this framework.

1.3.2 Our Cryptographic Analog of Self-testing: Remote Operator Application with Verifiability (ROAV)

We then introduce a new notion called remote operator application with verifiability (ROAV), as our answer to the question in Section 1.2.2.

Let's first review how the self-testing-based protocols typically work in the non-local game setting. Suppose the verifier would like to make use of the prover 1 and prover 2 to achieve some tasks – - for example, testing the ground state energy of a local Hamiltonian. One typical technique is to design two subprotocols π_{test} and π_{comp}. Subprotocol π_{test} is a non-local game with a self-testing property, while subprotocol π_{comp} is to test the Hamiltonian. Furthermore, the games are designed specially so that the prover 2, without communicating with the prover 1, could not decide which subprotocol the verifier is currently performing. Then the setting, the overall protocol and the security proof roughly go as follows:

- (Setting) The prover 1 and prover 2 initially hold EPR states. They receive questions from the verifier, make the corresponding measurements and send back the results.
- (Overall protocol) The verifier randomly chooses to execute either π_{test} or π_{comp} (without telling the provers the choices).
- (Security proof)
  1. To pass the overall protocol the provers have to pass π_{test} with high probability. By the property of π_{test} the operations of the prover 2 has to be close to the honest behavior.
  2. By the design of π_{test} and π_{comp}, and the fact that the prover 2 is close to the honest behavior in π_{test}, we could argue that the prover 2 is also close to the honest behavior in π_{comp}.
  3. Since we already know the prover 2 is close to be honest in π_{comp}, it's typically easy to analyze the execution of π_{comp} directly and show that it achieves the task.

Recall that we would like to define a single-server cryptographic analog of the non-local game self-testing where the single server plays the role of one of the two provers. So what does the "non-local game self-testing" mean here? Our idea is to consider both the step 1 and step 2 in the security proof template above as the "non-local game self-testing". Then let's focus on the prover 2 (as "one of the two provers" in our question) and assume the prover 1 is honest. Then we could do the following simplifications in the non-local game self-testing:

- In π_{test} we could assume the prover 1 first measures its states following the verifier's question; as a result, the joint state of the prover 1 and prover 2 becomes a cq-state where the verifer knows its description. Then the verifier also gets the measurement results from the prover 1's answer; so in the end we only need to consider the joint cq-state between the verifier and the prover 2.
- For π_{comp}, since we do not consider the step 3 above as a part of the self-testing notion, the question to the prover 1 in π_{comp} could be left undetermined. Then π_{comp} is not designed for any specific task any more, which in turn makes the self-testing a general notion.

Now the setting and the first two steps in the security proof could be updated as follows:

- (Setting) In π_{test} the client and the prover 2 initially hold a cq-state, where the verifier holds the classical part and the prover 2 holds the quantum part. In π_{comp} the prover 1 and prover 2 hold EPR states and the verifier does not have access to the prover 1's information.
- (Security proof) To make the first two steps in the security proof template above go through, we should at least require that:
  "The prover could pass in π_{test}" should imply that the prover's operation in π_{comp} is close to the desired one.

This gives us the basic intuition for formalizing our single-server cryptographic analog of the non-local game self-testing. Below we introduce the notion, which we call *reomte operator application with verifiability* (ROAV).

An ROAV for a POVM $(\mathcal{E}_1, \mathcal{E}_2, \cdots, \mathcal{E}_D)$ is defined as a tuple $(\rho_{\text{test}}, \pi_{\text{test}}, \pi_{\text{comp}})$ where:

- The setting contains the following registers. $\boldsymbol{D}$ is a client-side classical register, $\boldsymbol{Q}$ is a server-side quantum register, $\boldsymbol{P}$ is a quantum register in the environment with the same dimension with $\boldsymbol{Q}$.
- ρ_{test} is a cq-state on registers $\boldsymbol{D}$ and $\boldsymbol{Q}$.
- π_{test} is the protocol for the test mode. In the test mode ρ_{test} is used as the input state and $\boldsymbol{P}$ is empty.
- π_{comp} is the protocol for the computation mode (where the operations are applied). Denote the state of maximal entanglement (multiple EPR pairs) between $\boldsymbol{P}$ and $\boldsymbol{Q}$ as Φ. In the comp mode Φ is used as the input state and $\boldsymbol{D}$ is empty. Note that the execution of π_{comp} does not touch $\boldsymbol{P}$.

The soundness of ROAV is defined roughly as follows: for any adversary Adv, at least one of the following is true:

- In the test mode (π_{test} is executed with input state ρ_{test} against adversary Adv), the adversary gets caught cheating with significant probability;
- In the comp mode (π_{comp} is executed with input state Φ against adversary Adv), the final state gives the outcome of the following operations:
  The measurement described by $(\mathcal{E}_1, \mathcal{E}_2, \cdots, \mathcal{E}_D)$ is applied on $\boldsymbol{Q}$, and the client gets the measurement result $i \in [D]$; the corresponding output state (on register $\boldsymbol{P}$ and $\boldsymbol{Q}$) is $(\mathbb{I} \otimes \mathcal{E}_i)(\Phi)$.

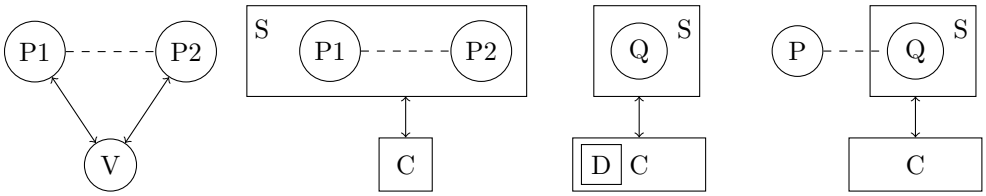

Figure 1 From the left to the right: the non-local game self-testing, previous definitions of its single party analog, and our definition (π_{test} and π_{comp} for the last two diagrams). Here $\leftrightarrow$ stands for interactions, and --- stands for quantum entanglements; C stands for the client and S stands for the server.

We finally note that the definition of ROAV also use the simulation-based security definition paradigm, like RSPV.

ROAV as the operator analog of RSPV

In the previous discussion we focus on the analog between our ROAV primitive and the non-local game self-testing. There is also another intuition that is analogous to the intuition of RSPV (see Section 1.1.1): In the ROAV problem the server is provided an undetermined input state, and the client would like to apply an operation (from an operator family) on it; in the end the client knows the description of the operation, while the server simply holds the output of the operation.

We also note that such a state-operator analog also appears in other directions like [36, 23].

Abstract constructions using ROAV

After giving the abstract definitions, we show several potential applications of our notions. We first show that ROAV is potentially a useful tool for constructing RSPV protocols for more general state families. Then we construct a Hamiltonian ground energy testing protocol based on specific RSPV and ROAV. Our construction shares similarities to Grilo's Hamiltonian verification protocol in the 2-party setting [21]. We note that these constructions are abstract constructions and concrete constructions for nontrivial ROAV remain open.

1.3.3 New RSPV Constructions

Now we introduce a series of new RSPV constructions. Our results not only give arguably simplified constructions for existing results but also cover new state families.

Overall approach

Instead of constructing each protocol directly from cryptographic assumptions, we study how different protocols could be *reduced* to existing ones in a black-box manner. Following this approach, we build a series of RSPV protocols step by step from RSPV for BB84 states $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$. Furthermore, these steps could be classified into two classes:

- In one class of steps, the reduction either has a simple intuition or is an application of the abstract properties in our framework (see Section 1.3.1) (in more detail, sequential composability and amplification).

- In the other class of steps, we could work on an "information-theoretic core" (IT-core) where the analysis is purely quantum information theoretic, and there is no appearance of computational notions like computational indistinguishability.

Our reductions are illustrated in Figure 2. We elaborate details in Section 2. As an example, let's describe the first step of our reductions. We assume an RSPV for BB84 states; then the client would like to prepare a sequence of such states in which there is only one $|+\rangle$ state and no $|-\rangle$ state. Equivalently, this could be written as $\frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle)$ where the hamming weight of $x_0 \oplus x_1$ is exactly 1. What the client needs to do is to repeat the RSPV-for-BB84 protocol for many rounds and tell the server which states it wants to keep. This could be easily understood as a repeat-and-pick process.

We argue that our approach is cleaner and easier to understand compared to the original construction in [20] at least in the following sense: In [20] the computational indistinguishability arguments and quantum information theoretic arguments are mixed together, which could lead to complicated details [1]. By separating two types of steps in the reductions explicitly, each step has a relatively simple intuition and there is much less room for complicated details.

Main results

Among these reductions, we consider the following two results particularly interesting.

▶ **Theorem 1.** *Assuming the existence of RSPV for BB84 states, there exists an RSPV for state family* $\{\frac{1}{\sqrt{2}}(|0\rangle |x_0\rangle + |1\rangle |x_1\rangle), x_0, x_1 \in \{0,1\}^n\}$.

This is an RSPV protocol for a new state family. Note that although there are plenty of works [10, 12] that allow the client to prepare these states with some other types of security (like claw-freeness, etc), as far as we know, this is the first time that an RSPV for it is constructed.

We also recover the results of RSPV for 8-basis states [20].

▶ **Theorem 2.** *Assuming the existence of RSPV for BB84 states, there exists an RSPV for state family* $\{|+_\theta\rangle := \frac{1}{\sqrt{2}}(|0\rangle + e^{\mathrm{i}\pi\theta/4}|1\rangle), \theta \in \{0,1,2\cdots 7\}\}$.

Instantiation of RSPV for BB84

We note that we still need to instantiate the RSPV for BB84 part to get a concrete protocol. Luckily the RSPV for BB84 has been studied relatively thoroughly: there are multiple constructions [19, 11] and we have a better understanding on the assumptions needed [11]. After instantiating the BB84 part by [11], we get RSPV constructions for these state families from weak NTCF, without requiring the adaptive hardcore bit property (elaborated below).

1.3.4 Application: CVQC From Cryptographic Group Actions

Now we apply our results to the classical verification of quantum computations (CVQC) problem.

As the preparation, we give a more detailed review on the variants of NTCF, and their relations to RSPV. We refer to Section 1.2.4 for a CVQC-centric background.

More backgrounds

Noisy trapdoor claw-free functions (NTCF) is a popular and powerful primitive in quantum cryptographic tasks like CVQC and RSPV. Informally, this primitive is defined to be a function family that satisfies the following requirements. Below we use f to denote a function sampled from this function family.

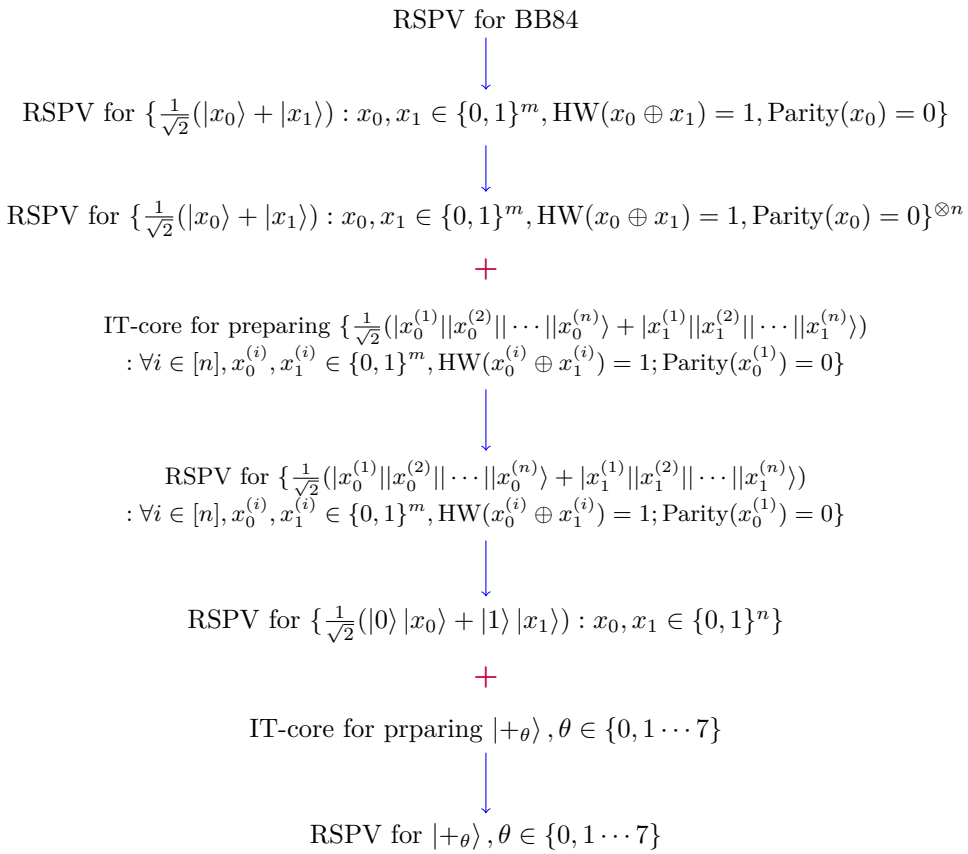

Figure 2 The reduction diagram. Here $\rightarrow$ means this step either has a simple intuition or is from the framework that we describe in Section 1.3.1; $+$ means the analysis of the IT-core below it is purely quantum information theoretic, and the RSPV above it is used to compile the IT-core to a full cryptographic protocol. HW is the hamming weight, Parity is the total parity.

- Trapdoor: this means that all the parties could evaluate f, but only the client, who holds the "trapdoor" information, could invert f.
- Noisy 2-to-1: the un-noisy 2-to-1 means that for each y in the range there exist exactly two preimages x_0, x_1 such that $f(x_0) = f(x_1) = y$. "Noisy" means that the evaluation of the function could be randomized, which makes the instantiation of the primitive easier. Below when we describe other properties we use the un-noisy version to simplify the description.
- Claw-free: the adversary could not efficiently find x_0, x_1 such that $f(x_0) = f(x_1)$.

In practice, we often use some variants of NTCF instead of the standard requirements above, to either make it more powerful or make the instantiation easier. Popular variants or additional requirements of NTCF include:

- Adaptive hardcore bit: the adversary could not find (d, y) with probability better than $\frac{1}{2}$ (the probability of random guessing) such that $d \cdot x_0 \equiv d \cdot x_1 \mod 2 \wedge d \neq 0$, where x_0, x_1 are two preimages of y.
  The RSPV for BB84 states is known to exist without this property [11], but Mahadev's constructions for CVQC [27] and RSPV for $|+_\theta\rangle$ [20] require this property.
- Extended function family: there is another function family g that is injective (instead of 2-to-1) and indistinguishable to f. Existing constructions for CVQC and RSPV for $|+_\theta\rangle$ [27, 20] also require this property.

- Inverse-polynomial correctness error (or weak correctness): this means that the 2-to-1 property is allowed to hold up to an inverse-polynomial error. This is used in [5] to define a primitive called weak TCF.

In our work we build our protocols on weak NTCF, that is, we allow inverse-polynomial correctness error and do not require additional properties like the adaptive hardcore bit property. In other words, we only use a relatively weak assumption from different variants of NTCF/TCF. This assumption could be instantiated from either LWE [10] or assumptions on group actions [5].

Applications of our results on CVQC

We first note that, [11] could be adapted easily to weak NTCF:

▶ **Theorem 3** (By [11]). *There exists an RSPV for BB84 assuming weak NTCF.*

As discussed in Section 1.3.3, this gives us a series of RSPV protocols from weak NTCF.

Then by [17], if the client samples and sends a series of $|+_\theta\rangle$ states to the server, it could do quantum computation verification using these states. Combining Fact 3, Theorem 2 and [17], we get:

▶ **Theorem 4.** *Assuming the existence of weak NTCF, there exists a classical verification of quantum computation protocol.*

Finally we recall the results in [5]:

▶ **Theorem 5** ([5]). *Under certain assumptions on cryptographic group actions, there exists a family of weak TCF.*

Thus we have:

▶ **Theorem 6.** *Under certain assumptions on cryptographic group actions, there exists a CVQC protocol.*

As an additional note, the work in [5] is largely on how to deal with the adaptive hardcore bit property; the fact that we do not need the adaptive hardcore bit may help us simplify the analysis or even construction in [5].

1.4 More Related Works

Previous versions

Compare to the previous versions, the abstract framework parts (described in Section 1.3.1, 1.3.2) of this work are sigfinicantly updated, and the concrete constructions part (described in Section 1.3.3, 1.3.4) are completely new.

RSP with other types of security

There are many works about remote state preparation but with other types of security (instead of RSPV). These works may or may not use the name "remote state preparation". As examples, [22] gives an RSP for the gadgets in [16]; [37] gives an RSP for the quantum money states. When we only consider the honest behavior, RSP is a very general notion.

Other state-operation analog

There are also several notions in quantum cryptography and complexity theory that have a state-operation analog. As examples, [36] studies the complexity of interactive synthesis of states and unitaries. [23] studies pseudorandom states and unitaries. In a sense, the relation of states and unitaries in these works is a "state-operation analog", as the RSPV and ROAV in our work.

1.5 Open Questions and Summary

Our work gives rise to a series of open questions; we consider the following two particularly interesting.

- One obvious open question coming out of this work is to give a construction for ROAV. Our work focuses on its definitions and applications in an abstract sense; an explicit construction of ROAV would allow us to instantiate these applications.
- For RSPV, although our results give new constructions for new state families, this is still far away from a general solution. Ideally we would like to have an RSPV for each computationally efficient state family. Whether this is possible and how to achieve it remain open.

In summary, two major part of this work is the framework and the constructions of RSPV protocols. By formulating and choosing notions and studying their abstract properties, we build an abstract framework for RSPV that is sufficiently well-behaved, which enables us to build more advanced, complicated protocols from more elementary, easy-to-construct components. Building on this framework, we construct a series of new RSPV protocols that not only (in a sense) simplify existing results but also cover new state families. Then we combine our results with existing works to show that the CVQC problem could be constructed from assumptions on cryptographic group actions. We also raised a new notion for certifying the server's operations. We consider our results as a solid progress in the understanding of RSPV; hopefully this will lay the foundation for further works.

2 Overview for Concrete Constructions of RSPV Protocols

We refer to the full version for the details of our contributions. In this section we elaborate our concrete RSPV constructions. As a preparation, we explain several basic notions or theorems in our framework.

- Sequential repetition: it roughly means that if protocol π_1 is an RSPV for state family $\mathcal{F}_1$, protocol π_2 is an RSPV for state family $\mathcal{F}_2$, then the sequential composition of π_1 and π_2 is an RSPV for state family $\mathcal{F}_1 \otimes \mathcal{F}_2$. This implies that, given an RSPV for some state family, its n-fold sequential composition is an RSPV for the n-fold tensor product of the state family.
- RSPV as a protocol compiler: it roughly means that, given a protocol starting with "the client prepares and sends some quantum states", we could replace this step with an RSPV protocol for these states, and the security property of the overall protocol will be preserved (approximately).
- PreRSPV and its amplification to RSPV:
  The preRSPV is defined as a pair of protocols $(\pi_{\text{test}}, \pi_{\text{comp}})$ which could be seen as a generalization of RSPV; it contains a test-mode protocol and a comp-mode protocol. The soundness is defined intuitively as follows: if π_{test} running against an adversary Adv could pass with high probability (that is, close to 1), then π_{comp} running against the same adversary Adv will satisfy the RSPV soundness.

Then a preRSPV defined above could be amplified to an RSPV by a *cut-and-choose* procedure: for each round, the client randomly chooses to execute either π_{test} or π_{comp} with some probabilities (without telling the server the choices). If the server could keep passing in many rounds, then for most of the π_{comp} rounds the target state is prepared as expected.

- PreRSPV-with-score and its amplification to RSPV:
  We also formalize a variant of preRSPV where the client counts *scores* for the server besides the pass/fail flag. The set-up is as follows: in the end of the protocol the client will record a win/lose decision in a ***score*** register, and the honest server could achieve the optimal winning probability. We use OPT to denote the optimal winning probability. (Note that when OPT = 1 the notion downgrades to the original preRSPV notion without the score.) The soundness is roughly defined as follows: if π_{test} running against an adversary Adv could win with probability close to OPT (and also pass with probability close to 1), then π_{comp} running against the same adversary Adv will satisfy the RSPV soundness.
  In the amplification procedure, the client needs to count the number of winning rounds and see whether the total score is close to the expected value of an honest execution.

Below we elaborate each step in the reduction diagram (Figure 2).

2.1 From BB84 to OneBlock and OneBlockTensor

We first build an RSPV for state family

$$\{\frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle : x_0, x_1 \in \{0,1\}^m, \text{HW}(x_0 \oplus x_1) = 1, \text{Parity}(x_0) = 0)\}$$

from the RSPV for BB84 states; this protocol is called OneBlock. The intuition is as described in Section 1.3.3: notice that states in this family could be written as sequence of BB84 states where there is only one $|+\rangle$ state and no $|-\rangle$ state; thus the client only needs to do a repeat-and-pick on RSPV-for-BB84 protocol.

Then we build the RSPV for

$$\{\frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle) : x_0, x_1 \in \{0,1\}^m, \text{HW}(x_0 \oplus x_1) = 1, \text{Parity}(x_0) = 0\}^{\otimes n} \tag{1}$$

which is the tensor products of the state family in OneBlock; this protocol is called OneBlockTensor. This is by taking the sequential repetition of the protocol OneBlock.

2.2 Construction of MultiBlock

Then we go from OneBlockTensor to construct an RSPV for the following state family:

$$\{\frac{1}{\sqrt{2}}(|x_0^{(1)}||x_0^{(2)}||\cdots||x_0^{(n)}\rangle + |x_1^{(1)}||x_1^{(2)}||\cdots||x_1^{(n)}\rangle) :$$

$$\forall i \in [n], x_0^{(i)}, x_1^{(i)} \in \{0,1\}^m, \text{HW}(x_0^{(i)} \oplus x_1^{(i)}) = 1; \text{Parity}(x_0^{(1)}) = 0\} \tag{2}$$

Roughly speaking, the client forces the server to measure the state in (1) to get (2). To see this clearly, we could first consider two blocks as an example. Suppose the server initially holds the state

$$(|x_0^{(1)}\rangle + |x_1^{(1)}\rangle) \otimes (|x_0^{(2)}\rangle + |x_1^{(2)}\rangle), \quad \forall b, i, \text{Parity}(x_b^{(i)}) = b$$

The client asks the server to measure the total parity of the strings it holds. Then if the server performs the measurement honestly, the state will collapse to:

$$\text{outcome} = 0 : |x_0^{(1)}||x_0^{(2)}\rangle + |x_1^{(1)}||x_1^{(2)}\rangle \tag{3}$$

$$\text{outcome} = 1 : |x_0^{(1)}||x_1^{(2)}\rangle + |x_1^{(1)}||x_0^{(2)}\rangle \tag{4}$$

The client could update the keys (that is, the state description) using the reported outcome and the original keys. If the client does the same for each $i = 2, 3 \cdots n$, in the honest setting the state in (1) will collapse to (2).

So what if the server cheats? One possible attack is that the server may not do the measurements and report the total parity honestly. To detect this attack, we will first construct a preRSPV (MultiBlockTest, MultiBlockComp) and use MultiBlockTest to test the server's behavior. In MultiBlockTest after getting the total parities the client will ask the server to measure all the states on the computational basis and report the measurement results; the client could check the results with its keys and see whether the results is consistent with the original keys and the total parities. As a concrete example, in (3) if the client asks the server to measure all the states, the measurement results should be either $x_0^{(1)}||x_0^{(2)}$ or $x_1^{(1)}||x_1^{(2)}$, otherwise the server is caught cheating.

After we get a preRSPV, we could amplify it to get an RSPV for (2).

Let's briefly discuss how the security proof goes through. One desirable property of the security proof of this step is that, we only need to analyze a "information-theoretic core": if we assume the initial state is (1), the analysis of the preRSPV will be purely information-theoretic, which means, the soundness holds against unbounded provers and we do not need to work on computational notions in the security analysis. After we prove the soundness of this information-theoretic part, we could prove the soundness of the overall protocol by calling the abstract properties in our framework.

2.3 Construction of KP

Then we go from MultiBlock to construct an RSPV for state family

$$\{\frac{1}{\sqrt{2}}(|0\rangle |x_0\rangle + |1\rangle |x_1\rangle) : x_0, x_1 \in \{0,1\}^n\} \tag{5}$$

This is by first calling MultiBlock to prepare a sufficiently big state in the form of (2), and then letting the client reveals suitable information to allow the server to transform (2) to (5). Let's explain the intuition.

We first note that by calculating the parity of the first block, (2) could be transformed to

$$\frac{1}{\sqrt{2}}(|0\rangle |x_0^{(1)}||x_0^{(2)}||\cdots\rangle + |1\rangle |x_1^{(1)}||x_1^{(2)}||\cdots\rangle) : \forall i, x_0^{(i)}, x_1^{(i)} \in \{0,1\}^m, \mathrm{HW}(x_0^{(i)} \oplus x_1^{(i)}) = 1 \tag{6}$$

The difference to (5) is that these keys are not sampled uniformly randomly. Note that in (6) we omit some conditions on the keys and focus on the most significant one.

Then the client will reveal lots of information about these keys, which allows the server to transform each two blocks in (6) to a pair of uniform random bits. Let's use the first two blocks as an example. The client will reveal $x_0^{(1)}$ and $x_1^{(2)}$. Then by doing xor between them and the corresponding blocks, (6) becomes:

$$\frac{1}{\sqrt{2}}(|0\rangle |0^m||000\cdots 1000\cdots||\cdots\rangle + |1\rangle |000\cdots 1000\cdots||0^m||\cdots\rangle) \tag{7}$$

Now the $000\cdots1000\cdots$ could be seen as a unary encoding of a random number in $[m]$. By choosing m to be power of 2, converting unary encoding to binary encoding and trimming out extra zeros, this state becomes

$$\frac{1}{\sqrt{2}}(|0\rangle|\gamma_0^{(1)}||\cdots\rangle + |1\rangle|\gamma_1^{(1)}||\cdots\rangle) \tag{8}$$

where $\gamma_0, \gamma_1 \in \{0,1\}^{[\log m]}$ and are uniformly random. Doing this for each two blocks in (6) gives (5).

2.4 Construction of QFac (RSPV for $|+_\theta\rangle$)

Now we have an RSPV for states $\frac{1}{\sqrt{2}}(|0\rangle|x_0\rangle + |1\rangle|x_1\rangle)$ with uniformly random x_0, x_1; we are going to construct an RSPV for the state[1]

$$|+_\theta\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{\mathrm{i}\pi\theta/4}|1\rangle), \theta \in \{0,1,2\cdots7\}.$$

We first note that existing work [20] also takes a similar approach: the client first instruct the server to prepare the state $|0\rangle|x_0\rangle + |1\rangle|x_1\rangle$, and then transforms it to $|+_\theta\rangle$.

Let's explain the constructions. The overall ideas for the construction are basically from [20] (adapted to the languages of our framework). We will first construct a preRSPV-with-score, as follows: in both the test mode and comp mode the client will instruct the server to prepare $|+_\theta\rangle$ state, then in the test mode the client will instruct the server to measure $|+_\theta\rangle$, and stores a score based on the reported result; in the comp mode $|+_\theta\rangle$ will be kept. Then once we show this protocol is indeed a preRSPV, we could amplify it to an RSPV as given in our framework.

The first step is to allow the honest server to transform $\frac{1}{\sqrt{2}}(|0\rangle|x_0\rangle + |1\rangle|x_1\rangle)$ to $|+_\theta\rangle$. There are multiple ways to do it, for example:

1. The client will first instruct the server to introduce a phase of $e^{\mathrm{i}\pi\theta_{2,3}/4}, \theta_{2,3} \in \{0,1,2,3\}$ where $\theta_{2,3}$ are hidden in the server's view. This could be done by selecting the xor of the first two bits of x_0, x_1 as θ_2, θ_3. Then on the one hand $\theta_{2,3} = 2\theta_2 + \theta_3$ will be completely hidden; on the other hand using the phase-table-like technique in [39, 41] the honest server could introduce a phase of $e^{\mathrm{i}\pi\theta_{2,3}/4}$ to the x_1 branch up to a global phase.
2. The server does a Hadamard measurement on each bit of the x-part and get a measurement result d; this introduces a phase of $e^{\mathrm{i}\pi(d\cdot(x_0+x_1))}$ to the qubits. Then the server sends back d to the client and the client could calculate $\theta_1 = (d\cdot(x_0+x_1)) \mod 2$. The server holds a single qubit in the state $|+_\theta\rangle, \theta = 4\theta_1 + 2\theta_2 + \theta_3$.

Now in the comp mode we are done. In the test mode the client will continue to ask the server to make measurement on a random basis $|+_\varphi\rangle, |+_{\varphi+4}\rangle, \varphi \leftarrow_r \{0,1,2\cdots7\}$. First we could see that when $\theta = \varphi$ the measurement will collapse to $|+_\varphi\rangle$ with probability 1, and when $\theta = \varphi + 4$ the measurement will collapse to $|+_{\varphi+4}\rangle$ with probability 1. Thus the client could record a "win" score if he has seen such an outcome. But solely doing this does not give us a full control on the server's behavior and states; an important idea is that, when φ is close to θ (or $\theta + 4$), the measurement should also collapse to $|+_\varphi\rangle$ (or $|+_{\varphi+4}\rangle$, correspondingly). This gives some probability of losing even for an honest server; however by analyzing the game it's possible to say "if the server wins with probability close to the optimal winning probability, the state before the testing measurement has to be close to the target state up to an isometry", which is still sufficient for amplification.

[1] The protocol name QFac is from [15].

Security proofs, existing works and their limitations, and our approach

So how could the quoted claim just now be proved? Existing works like [20, 15] has already done a lot of works on this part. In [15] the authors introduce a notion called *blind self-testing*. In more detail, let's denote the server-side state by the time that the comp mode is done corresponding to the client-side phase θ as ρ_θ. (In other words, the overall state is $\sum_\theta |\theta\rangle\langle\theta| \otimes \rho_\theta$.) Then the blind self-testing requires that the state $\rho_{\theta_{2,3}} + \rho_{\theta_{2,3}+4}$ is the same for any $\theta_{2,3}$, which is called (information-theoretic) *basis-blindness* (here $\theta_{2,3}$ is the "basis" and the notion means that the basis is completely hidden after randomization of θ_1). [15] shows that if the initial state satisfies the basis blindness property, the claim "high winning probability $\Rightarrow$ close to the target state up to an isometry" holds.

However, the proof of this claim given in [15] does not generalize to the computational analog of basis blindness. [15] does not solve the problem and leave the security of the whole protocol as a conjecture. [20] makes use of computational indistinguishability between states in the form of $\rho_{\theta_{2,3}} + \rho_{\theta_{2,3}+4}$ together with some quantum information theoretic arguments to prove the claim; in their proofs computational indistinguishability and quantum information theoretic arguments are mixed together, which could be complicated to work on and lead to sophisticated details [1].

Our new hammer for getting rid of this problem is the KP protocol, which is an RSPV for $|0\rangle|x_0\rangle + |1\rangle|x_1\rangle$. (Note that in previous works the preparation of $|0\rangle|x_0\rangle + |1\rangle|x_1\rangle$ is not known to satisfy the RSPV soundness in the malicious setting.) By starting from KP, we are able to prove the security in a much nicer way:

1. The "information-theoretic core": in the security proof we could first simply assume the server holds exactly the state $|0\rangle|x_0\rangle + |1\rangle|x_1\rangle$; then we could prove the ρ_θ generated from it has the (information-theoretic) basis blindness property; then the analysis of the testing on $|+_\theta\rangle$ is basically from existing results [20, 15].
2. Once we complete the analysis of this information-theoretic core, we could compile it using results in our framework to an RSPV.

2.5 A Summary

In summary we have achieve each step of the reductions as shown in Figure 2. The construction of OneBlock is by a repeat-and-pick procedure, OneBlockTensor is by the sequential composition, KP is by revealing some information to allow the server to transform the state to some other forms. For constructions of MultiBlock and QFac, we first analyze an "information-theoretic core" where we only need to work on statistical closeness, and compile the IT-core (IT=information-theoretic) to a full protocol by calling existing soundness properties.

References

1 Discussion with Thomas Vidick on details of paper "computationally secure and composable remote state preparation", 2024.

2 Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive proofs for quantum computations. In Andrew Chi-Chih Yao, editor, *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 453–469. Tsinghua University Press, 2010. URL: `http://conference.iiis.tsinghua.edu.cn/ICS2010/content/papers/35.html`.

3 Gorjan Alagic, Andrew M. Childs, Alex B. Grilo, and Shih-Han Hung. Non-interactive classical verification of quantum computation. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part III*, volume 12552 of *Lecture Notes in Computer Science*, pages 153–180. Springer, 2020. doi:10.1007/978-3-030-64381-2_6.

4 Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 411–439, Berlin, Heidelberg, 2020. Springer. doi:10.1007/978-3-030-64834-3_14.

5 Navid Alamati, Giulio Malavolta, and Ahmadreza Rahimi. Candidate trapdoor claw-free functions from group actions with applications to quantum protocols. In Eike Kiltz and Vinod Vaikuntanathan, editors, *Theory of Cryptography - 20th International Conference, TCC 2022, Chicago, IL, USA, November 7-10, 2022, Proceedings, Part I*, volume 13747 of *Lecture Notes in Computer Science*, pages 266–293, Berlin, Heidelberg, 2022. Springer. doi:10.1007/978-3-031-22318-1_10.

6 Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019. doi:10.1038/s41586-019-1666-5.

7 James Bartusek. Secure quantum computation with classical communication. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part I*, volume 13042 of *Lecture Notes in Computer Science*, pages 1–30, Cham, 2021. Springer. doi:10.1007/978-3-030-90459-3_1.

8 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, July 2001. doi:10.1103/PhysRevLett.87.077902.

9 Dolev Bluvstein et al. A quantum processor based on coherent transport of entangled atom arrays. *Nature*, 604(7906):451–456, 2022. doi:10.1038/s41586-022-04592-6.

10 Zvika Brakerski, Paul F. Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 320–331. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00038.

11 Zvika Brakerski, Alexandru Gheorghiu, Gregory D. Kahanamoku-Meyer, Eitan Porat, and Thomas Vidick. Simple tests of quantumness also certify qubits. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part V*, volume 14085 of *Lecture Notes in Computer Science*, pages 162–191. Springer, 2023. doi:10.1007/978-3-031-38554-4_6.

12 Zvika Brakerski, Venkata Koppula, Umesh V. Vazirani, and Thomas Vidick. Simpler proofs of quantumness. In Steven T. Flammia, editor, *15th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2020, June 9-12, 2020, Riga, Latvia*, volume 158 of *LIPIcs*, pages 8:1–8:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, May 2020. doi:10.4230/LIPIcs.TQC.2020.8.

13 Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 395–427, Berlin, Heidelberg, 2018. Springer. doi:10.1007/978-3-030-03332-3_15.

14 Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. Classical verification of quantum computations with efficient verifier. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part III*, volume 12552 of *Lecture Notes in Computer Science*, pages 181–206, Cham, 2020. Springer. doi:10.1007/978-3-030-64381-2_7.

15 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. doi:10.1007/978-3-030-34578-5_22.

16 Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum homomorphic encryption for polynomial-sized circuits. *IACR Cryptol. ePrint Arch.*, 2016:559, 2016. URL: http://eprint.iacr.org/2016/559.

17 Samuele Ferracin, Theodoros Kapourniotis, and Animesh Datta. Reducing resources for verification of quantum computations. *Phys. Rev. A*, 98:022323, August 2018. doi:10.1103/PhysRevA.98.022323.

18 Honghao Fu, Daochen Wang, and Qi Zhao. Parallel self-testing of EPR pairs under computational assumptions. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*, volume 261 of *LIPIcs*, pages 64:1–64:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ICALP.2023.64.

19 Alexandru Gheorghiu, Tony Metger, and Alexander Poremba. Quantum cryptography with classical communication: parallel remote state preparation for copy-protection, verification, and more. *IACR Cryptol. ePrint Arch.*, page 122, 2022. URL: https://eprint.iacr.org/2022/122.

20 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. doi:10.1109/FOCS.2019.00066.

21 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, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 28:1–28:13, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2019.28.

22 Aparna Gupte and Vinod Vaikuntanathan. How to construct quantum fhe, generically. *IACR Cryptol. ePrint Arch.*, page 893, June 2024. URL: https://eprint.iacr.org/2024/893, arXiv:2406.03379.

23 Zhengfeng Ji, Yi-Kai Liu, and Fang Song. Pseudorandom quantum states. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 126–152, Cham, 2018. Springer International Publishing. doi:10.1007/978-3-319-96878-0_5.

24 Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. Mip* = RE. *Commun. ACM*, 64(11):131–138, October 2021. doi:10.1145/3485628.

25 Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Lisa Yang. Quantum advantage from any non-local game. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, STOC 2023, pages 1617–1628, New York, NY, USA, 2023. ACM. doi:10.1145/3564246.3585164.

26 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. doi:10.1109/FOCS.2018.00039.

27 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. doi:10.1109/FOCS.2018.00033.

28 Dominic Mayers and Andrew Chi-Chih Yao. Self testing quantum apparatus. *Quantum Inf. Comput.*, 4(4):273–286, July 2004. doi:10.26421/QIC4.4-3.

29 Tony Metger and Thomas Vidick. Self-testing of a single quantum device under computational assumptions. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 19:1–19:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ITCS.2021.19.

30 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. doi:10.1103/PhysRevA.106.L010601.

31 Anand Natarajan and Tina Zhang. Bounding the quantum value of compiled nonlocal games: From CHSH to BQP verification. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 1342–1348. IEEE, 2023. doi:10.1109/FOCS57990.2023.00081.

32 Arun Kumar Pati. Minimum cbits required to transmit a qubit. *Phys. Rev. A*, 63:014320, 2001. doi:10.1103/PhysRevA.63.014320.

33 Sandu Popescu and Daniel Rohrlich. Which states violate bell's inequality maximally? *Physics Letters A*, 169(6):411–414, 1992. doi:10.1016/0375-9601(92)90819-8.

34 Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), September 2009. doi:10.1145/1568318.1568324.

35 Ben W. Reichardt, Falk Unger, and Umesh V. Vazirani. Classical command of quantum systems. *Nat.*, 496(7446):456–460, 2013. doi:10.1038/nature12035.

36 Gregory Rosenthal and Henry Yuen. Interactive proofs for synthesizing quantum states and unitaries. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 112:1–112:4. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ITCS.2022.112.

37 Omri Shmueli. Public-key quantum money with a classical bank. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, STOC 2022, pages 790–803, New York, NY, USA, 2022. ACM. doi:10.1145/3519935.3519952.

38 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. doi:10.1007/BF01207366.

39 Jiayu Zhang. Delegating quantum computation in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 30–60, Cham, 2019. Springer. doi:10.1007/978-3-030-36033-7_2.

40 Jiayu Zhang. Succinct blind quantum computation using a random oracle. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1370–1383, New York, NY, USA, 2021. ACM. doi:10.1145/3406325.3451082.

41 Jiayu Zhang. Classical verification of quantum computations in linear time. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 46–57, Los Alamitos, CA, USA, November 2022. IEEE. doi:10.1109/FOCS54457.2022.00012.

42 Qingling Zhu, Sirui Cao, Fusheng Chen, Ming-Cheng Chen, Xiawei Chen, Tung Hsun Chung, Hui Deng, Yajie Du, Daojin Fan, Ming Gong, Cheng Guo, Chu Guo, Shaojun Guo, Lian-Chen Han, Linyin Hong, Heliang Huang, Yongheng Huo, Liping Li, Na Li, Shaowei Li, Yuan Yuan Li, Futian Liang, Chun Lin, Jin Lin, Haoran Qian, Dan Qiao, Hao Rong, Hong-Bo Su, Lihua Sun, Liangyuan Wang, Shiyu Wang, Dachao Wu, Yulin Wu, Yu Xu, Kai Yan, Weifeng Yang, Yang Yang, Yangsen Ye, Jian Hua Yin, Chong Ying, Jiale Yu, Chen Zha, Cha Zhang, Haibin Zhang, Kaili Zhang, Yiming Zhang, Han Zhao, You-Wei Zhao, Liang Zhou, Chaoyang Lu, Cheng-Zhi Peng, Xiaobo Zhu, and Jian-Wei Pan. Quantum computational advantage via 60-qubit 24-cycle random circuit sampling. *Science bulletin*, 67 3:240–245, 2021. URL: https://api.semanticscholar.org/CorpusID:237442167.