Quantum Cryptography with Classical Communication: Parallel Remote State Preparation for Copy-Protection, Verification, and More

Conference Paper

Author(s):
Gheorghiu, Alexandru; Metger, Tony; Poremba, Alexander

Publication date:
2023-07-05

Permanent link:
https://doi.org/10.3929/ethz-b-000626982

Originally published in:
Leibniz International Proceedings in Informatics (LIPIcs) 261, https://doi.org/10.4230/LIPIcs.ICALP.2023.67

Quantum Cryptography with Classical Communication: Parallel Remote State Preparation for Copy-Protection, Verification, and More

Alexandru Gheorghiu
Department of Computer Science and Engineering, Chalmers University of Technology, Göteborg, Sweden
Institute for Theoretical Studies, ETH Zürich, Switzerland

Tony Metger
Institute for Theoretical Physics, ETH Zürich, Switzerland

Alexander Poremba
Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA

Abstract

Quantum mechanical effects have enabled the construction of cryptographic primitives that are impossible classically. For example, quantum copy-protection allows for a program to be encoded in a quantum state in such a way that the program can be evaluated, but not copied. Many of these cryptographic primitives are two-party protocols, where one party, Bob, has full quantum computational capabilities, and the other party, Alice, is only required to send random BB84 states to Bob. In this work, we show how such protocols can *generically* be converted to ones where Alice is fully classical, assuming that Bob cannot efficiently solve the LWE problem. In particular, this means that all communication between (classical) Alice and (quantum) Bob is classical, yet they can still make use of cryptographic primitives that would be impossible if both parties were classical. We apply this conversion procedure to obtain quantum cryptographic protocols with classical communication for unclonable encryption, copy-protection, computing on encrypted data, and verifiable blind delegated computation.

The key technical ingredient for our result is a protocol for *classically-instructed parallel remote state preparation of BB84 states.* This is a multi-round protocol between (classical) Alice and (quantum polynomial-time) Bob that allows Alice to certify that Bob must have prepared n uniformly random BB84 states (up to a change of basis on his space). While previous approaches could only certify one- or two-qubit states, our protocol allows for the certification of an n-fold tensor product of BB84 states. Furthermore, Alice knows which specific BB84 states Bob has prepared, while Bob himself does not. Hence, the situation at the end of this protocol is (almost) equivalent to one where Alice sent n random BB84 states to Bob. This allows us to replace the step of preparing and sending BB84 states in existing protocols by our remote-state preparation protocol in a generic and modular way.

2012 ACM Subject Classification Theory of computation; Theory of computation → Cryptographic protocols

Keywords and phrases Quantum cryptography, Remote state preparation, Self-testing, Learning with errors, Quantum copy-protection, Unclonable encryption, Quantum verification

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.67

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2201.13445`

Funding *Alexandru Gheorghiu*: Acknowledges support from the Knut and Alice Wallenberg Foundation through the Wallenberg Centre for Quantum Technology (WACQT) and Dr. Max Rössler, the Walter Haefner Foundation and the ETH Zürich Foundation, while at ETH.

50th International Colloquium on Automata, Languages, and Programming (ICALP 2023).
Editors: Kousha Etessami, Uriel Feige, and Gabriele Puppis; Article No. 67; pp. 67:1–67:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Tony Metger: Acknowledges support from the QuantERA project "eDict" and the Air Force Office of Scientific Research (AFOSR) Grant No. FA9550-19-1-0202.
Alexander Poremba: Acknowledges partial support from AFOSR YIP award number FA9550-16-1-0495 and the Institute for Quantum Information and Matter (an NSF Physics Frontiers Center; NSF Grant PHY-1733907)

Acknowledgements We thank Honghao Fu, Thomas Vidick, and Daochen Wang for helpful discussions, and Jeffrey Champion and John Wright for allowing us to use the results in Section 4.3 of the full version of the manuscript, which are based on unpublished joint work by them and the second author. We also thank Matty Hoban for pointing out a typo in an earlier draft.

1 Introduction

A central distinction between classical and quantum information is that a classical string can always be copied, but a quantum state cannot: the *no-cloning theorem* states that there cannot exist a procedure that produces the state $\rho \otimes \rho$ when given as input an arbitrary quantum state ρ [51]. The first cryptographic protocols that made use of the no-cloning theorem were Wiesner's proposal to use quantum states as unforgeable banknotes [50] and Bennett and Brassard's protocol for information-theoretically secure quantum key-distribution (the BB84 QKD protocol) [6]. These protocols rely on the idea of a *conjugate coding scheme*: classical information can be encoded into a quantum state in (at least) two incompatible bases, most commonly the standard basis $\{|0\rangle, |1\rangle\}$ and the Hadamard basis $\{|+\rangle, |-\rangle\}$, where $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$. These four states are commonly referred to as *BB84 states*. If we encode a bit $b \in \{0,1\}$ as either $|b\rangle$ or $|(-)^b\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^b|1\rangle)$, then an adversary who does not know which basis we chose for the encoding cannot create a copy of this quantum state. Furthermore, if the adversary tries to measure the state, with probability 1/2 they will choose the "wrong" measurement basis, which disturbs the state and means that the adversary's tampering can be detected.

There is an important conceptual difference between the BB84 protocol and Wiesner's quantum money scheme. The former addresses the problem of key-distribution, which is a task that can also be achieved classically under computational assumptions using public-key cryptography [18]. In contrast, Wiesner's quantum money scheme achieves a functionality which is entirely impossible classically, even under computational assumptions. Recently there has been renewed interest in this latter kind of application, i.e. to use BB84 states to construct quantum cryptographic primitives that have no classical analogue. Perhaps the most striking example of this is the idea of *quantum copy-protection* [1]. Suppose that a vendor has created a piece of software (viewed as a function that maps some input to some output) and wants to allow a user to run it (i.e. to evaluate the function), while preventing the user from producing additional "pirated" copies of the original software. Clearly, this is impossible classically: any piece of software is specified by a string of symbols, which can easily be copied. Surprisingly, it has been shown that it is possible to encode certain narrow classes of functions in the form of a quantum state in such a way that a user can evaluate the function without being able to copy it [16].

Copy-protection and many related protocols require only limited quantum capabilities from one party, e.g. the *vendor* in the case of copy-protection: they only need to prepare random BB84 states and send them to the other party (e.g. the *user* in copy-protection), who has full quantum computational capabilities. In particular, this requires a quantum channel between the two parties to send the BB84 states. The purpose of this paper is to show that such protocols, where one party's quantum operations are limited to preparing and sending random BB84 states, can be converted into protocols where that party is *fully classical*. This *dequantises* such protocols in the sense that all communication becomes classical. To achieve this, we need to construct a protocol between a *classical verifier* and a *computationally bounded quantum prover* that achieves the same outcome as if the verifier had prepared and sent random BB84 states to the prover. We call this task *classically-instructed parallel remote state preparation of BB84 states*, or *parallel RSP* for short. Our protocol builds on techniques introduced in [33, 7, 24] that allow the verifier to use post-quantum cryptography to constrain the actions of an untrusted (but computationally bounded) prover and certify the result of a certain computation or the preparation of certain states. In contrast to earlier works on remote state preparation (or *self-testing*) in this setting, which could only certify states comprised of a constant number of qubits, our protocol allows for the certification of an n-fold tensor product of states. We discuss the difference between our approach and previous approaches to RSP (in particular the protocol of [24]) in Section 4. Proving soundness for this parallel RSP protocol is the main technical result of our work. We then use this result to dequantise a number of cryptographic protocols, namely unclonable quantum encryption, quantum copy-protection, quantum computing on encrypted data and blind verification of quantum computation.

2 Main results

We start by first describing the soundness guarantee achieved by our parallel RSP protocol. Intuitively, the goal of our protocol is to guarantee that the prover has prepared a quantum state of the form $H^{\theta_1}|v_1\rangle\langle v_1|H^{\theta_1} \otimes \ldots \otimes H^{\theta_n}|v_n\rangle\langle v_n|H^{\theta_n}$, where $\vec{v}, \vec{\theta} \in \{0,1\}^n$. Additionally, the prover should not have any information about $\vec{v}$ and $\vec{\theta}$ beyond what is contained in its BB84 states, while the verifier should know both $\vec{v}$ and $\vec{\theta}$. Our protocol achieves a guarantee of this kind assuming the quantum-intractability of the *Learning with Errors* (LWE) problem introduced by Regev [43]. Our main result is the following (see the full manuscript for the corresponding formal statement):

▶ **Theorem 1** (Informal). *There exists an interactive protocol between a classical verifier and a computationally bounded quantum prover such that the following holds assuming the quantum-intractability of LWE (with quantum advice). Fix a number n of BB84 states. Consider any efficient prover strategy and let W and P be the verifier's and prover's systems at the end of the protocol, respectively. Then there exists an isometry $V : P \to QP'$ (for $\mathcal{H}_Q \cong (\mathbb{C}^2)^{\otimes n}$ and P' arbitrary) and an additional (subnormalised) state $\alpha_{P'}$ such that for any basis choice $\vec{\theta} \in \{0,1\}^n$, the protocol's final state σ_{WP} conditioned on the prover being accepted satisfies*

$$
\begin{aligned}
p_{\text{success}} V \sigma_{WP} V^\dagger \quad &\overset{c}{\approx}_{1/\operatorname{poly}(n)} \\
&\frac{1}{2^n} \sum_{\vec{v}\in\{0,1\}^n} |v\rangle\langle v|_W \otimes \left(H^{\theta_1}|v_1\rangle\langle v_1|H^{\theta_1} \otimes \ldots \otimes H^{\theta_n}|v_n\rangle\langle v_n|H^{\theta_n}\right) \otimes \alpha_{P'} \,.
\end{aligned}
$$

Here, p_{success} is the prover's success probability in the protocol and $\overset{c}{\approx}_{1/\operatorname{poly}(n)}$ denotes computational indistinguishability up to inverse polynomial error.

We make two remarks regarding this security guarantee. Firstly, the theorem makes a statement about the joint state of the verifier's system W and the prover's system P after applying an isometry V that *only acts on the prover's space*. This additional isometry is unavoidable: it represents the prover's freedom to use any basis of its choice on its space.

Hence, we cannot guarantee that the prover prepares BB84 states (in the standard basis), only that it prepares BB84 states up to a change of basis. However, crucially this change of basis is *independent of which BB84 state was supposed to be prepared*, i.e., V is independent of $\vec{v}$ and $\vec{\theta}$ (but it can of course depend on the prover's strategy). Put differently, the theorem guarantees that the prover prepares one of 4^n possible states whose relation to each other is the same as the relation between the 4^n BB84 states. This does not affect the utility of the prover's state for applications. In fact, this freedom also exists if the verifier sent n BB84 states to the prover via a quantum channel: the prover could apply an isometry V to these states immediately upon receipt, but the security of any application using the BB84 states is not impacted by this.

Secondly, the theorem holds *for any* basis choice $\vec{\theta}$, but *on average* over the values $\vec{v}$. In other words, in the protocol, the verifier gets to choose the bases at will, but the values will be uniformly random and cannot be chosen by the verifier. Furthermore, the only dependence on $\vec{v}$ and $\vec{\theta}$ in the prover's state is via the BB84 states. This means that the protocol forces the prover to prepare these states "blindly", i.e., the prover does not know which BB84 states were actually prepared. In contrast, the verifier does know, because they chose $\vec{\theta}$ and are in possession of the system W, which contains information about $\vec{v}$. This asymmetry of knowledge about the prover's state is the same as what is achieved by preparing and sending BB84 states through a quantum channel and is crucial for applications.

We also note that a consequence of Theorem 1 is the certification of an n-fold tensor product structure within the prover's system. This can be interpreted as saying that any successful prover must have a quantum memory capable of storing n-qubits. Being able to certify an n-qubit state in the prover's system is the main technical challenge towards proving soundness, as we outline in the next subsection. This notion of a computational proof of quantum space has been formalised in [21], who prove a similar parallel rigidity result to ours, but for a different class of states that does not immediately allow for cryptographic applications.

2.1 Soundness proof for parallel RSP protocol

The full RSP protocol is described as Protocol 3 (though our discussion here is restricted to Protocol 1). Its soundness proof can be found in the full version of the manuscript.

We briefly explain the difference between Protocol 1 and Protocol 3: Protocol 1 is a protocol to *test* the prover, i.e. in this protocol the prover is asked to prepare *and measure* a quantum state, and the verifier runs checks on the prover's answer. The soundness statement for this protocol is a self-testing statement in the sense of [36], which characterises which states and measurements the prover used in the protocol. Although we do not spell this out, it is easy to obtain an explicit self-testing statement from our proof. In contrast, Protocol 3 is a protocol for remote state preparation, so the prover is supposed to prepare, but not yet measure, a particular quantum state. Instead, this quantum state will be used for other applications. This means that we do not want to make a statement about how the prover measured its state, but rather what state remains in its quantum memory. The soundness of Protocol 3 follows from that of Protocol 1 via a statistical argument. In the following, we focus on Protocol 1. We do not explain the protocol and the cryptographic primitives underlying it in detail; instead, we give a very high-level description of the relevant part of the soundness proof of the RSP protocol from [24] and then explain our method for proving a parallel rigidity statement based on that result.

The main cryptographic primitive underlying the RSP protocol is a so-called extended noisy trapdoor claw-free function (ENTCF) family, which can be constructed assuming the quantum hardness of LWE [43, 33]. An ENTCF family is a family of functions indexed by

▶ **Protocol 1. Test round protocol.**

Let $\lambda \in \mathbb{N}$ be the security parameter, $(\mathcal{F}, \mathcal{G})$ an ENTCF family, and $n = \text{poly}(\lambda)$ the number of BB84 states that the verifier wishes to prepare.

1. The verifier selects a uniformly random basis $\theta \overset{\$}{\leftarrow} \{0,1\}$, where 0 corresponds to the computational and 1 to the Hadamard basis.
2. The verifier samples keys and trapdoors $(k_1, t_{k_1}; \dots k_n, t_{k_n})$ by computing $(k_i, t_{k_i}) \leftarrow \textsc{Gen}_{\mathcal{K}_\theta}(1^\lambda)$. The verifier then sends $(k_1, \dots k_n)$ to the prover (but keeps the trapdoors t_{k_i} private).
3. The verifier receives $(y_1, \dots, y_n) \in \mathcal{Y}^{\times n}$ from the prover.
4. The verifier selects a round type $\in$ {preimage round, Hadamard round} uniformly at random and sends the round type to the prover.
   a. For a *preimage round*: The verifier receives $(b_1, x_1; \ \dots b_n, x_n)$ from the prover, with $b_i \in \{0,1\}$ and $x_i \in \mathcal{X}$. The verifier sets $\texttt{flag} \leftarrow \texttt{fail}_{\texttt{Pre}}$ if $\textsc{Chk}(k_i, y_i, b_i, x_i) = 0$.
   b. For a *Hadamard round*: The verifier receives $d_1, \dots d_n \in \{0,1\}^w$ from the prover (for some w depending on the security parameter). The verifier sends $q = \theta$ to the prover, and receives answers $v_1, \dots v_n \in \{0,1\}$. The verifier performs the following checks:

Case	Verifier's check
$q = \theta = 0$	Set $\texttt{flag} \leftarrow \texttt{fail}_{\texttt{Had}}$ if $\hat{b}(k_i, y_i) \neq v_i$ for some i.
$q = \theta = 1$	Set $\texttt{flag} \leftarrow \texttt{fail}_{\texttt{Had}}$ if $\hat{u}(k_i, y_i, d_i) \neq v_i$.

Note. We denote the "question" separately by q (even though here we always have $q = \theta$) because when the variant of this protocol in Protocol 2 is used in the context of another cyrptographic task, the verifier can also send questions q which are different from θ.

▶ **Protocol 2. Preparation round protocol.**

Let $\lambda \in \mathbb{N}$ be the security parameter, $(\mathcal{F}, \mathcal{G})$ an ENTCF family, and $n = \text{poly}(\lambda)$ the number of BB84 states that the verifier wishes to prepare.

1. The verifier selects bases $\vec{\theta} \overset{\$}{\leftarrow} \{0,1\}^n$, where 0 corresponds to the computational and 1 to the Hadamard basis.
2. The verifier samples keys and trapdoors $(k_1, t_{k_1}; \dots k_n, t_{k_n})$ by computing $(k_i, t_{k_i}) \leftarrow \textsc{Gen}_{\mathcal{K}_{\theta_i}}(1^\lambda)$. The verifier then sends $(k_1, \dots k_n)$ to the prover (but keeps the trapdoors t_{k_i} private).
3. The verifier receives $y_1, \dots y_n \in \mathcal{Y}$ from the prover.
4. The verifier sends "Hadamard round" to the prover as the round type.
5. The verifier receives $d_1, \dots d_n \in \{0,1\}^w$ from the prover (for some w depending on the security parameter). The verifier computes a string $\vec{v}$ according to

$$v_i = \begin{cases} \hat{b}(k_i, y_i) & \text{if } \theta_i = 0\,, \\ \hat{u}(k_i, y_i, d_i) & \text{if } \theta_i = 1\,. \end{cases}$$

▶ **Protocol 3. Multi-round protocol for preparation of BB84 states.**

Let $\lambda \in \mathbb{N}$ be the security parameter, $(\mathcal{F}, \mathcal{G})$ an ENTCF family, $n = \mathrm{poly}(\lambda)$ the number of BB84 states that the verifier wishes to prepare, $N = M^2$ the maximum number of test rounds (for $M \in \mathbb{N}$), and δ an error tolerance parameter. For $j \in [M]$ we denote by $B_j = \{(j-1)M+1, \ldots, jM\}$ the j-th "block" of M rounds.

1. The verifier (privately) samples $S \xleftarrow{\$} \{0, \ldots, M-1\}$ (the number of M-round blocks of test rounds that will be performed).
2. The verifier performs SM executions of Protocol 1 with the prover. The verifier aborts if *for any* $j \in [S]$, the fraction of rounds in B_j for which $\texttt{flag} = \texttt{fail}_{\texttt{Pre}}$ or $\texttt{flag} = \texttt{fail}_{\texttt{Had}}$ exceeds δ.
3. The verifier (privately) samples $R \xleftarrow{\$} [M]$ and executes Protocol 1 with the prover $R-1$ times. Then, the verifier executes Protocol 2 with the prover and records the basis choice $\vec{\theta}$ and the string $\vec{v}$ from that execution.

a set of keys $\mathcal{K}_0 \cup \mathcal{K}_1$. $\mathcal{K}_0$ and $\mathcal{K}_1$ are disjoint sets of keys with the property that given a $k \in \mathcal{K}_0 \cup \mathcal{K}_1$, it is computationally intractable to determine which set this key belongs to. See [33, Section 4] for further details on ENTCF families.

In the RSP protocol from [24], for a given *basis choice* $\theta \in \{0,1\}$ (where "0" corresponds to the computational and "1" to the Hadamard basis), the verifier samples a key $k \in \mathcal{K}_\theta$, alongside some trapdoor information t. The verifier sends k to the prover and keeps t private. The verifier and prover then interact classically; for us, the main point of interest is the last round of the protocol, i.e. the last message from the verifier to the prover and back. Let us denote the protocol's transcript up to the last round by $\texttt{ts}$. Before the last round, the remaining quantum state of an *honest* prover is the single-qubit state $H^\theta |v\rangle\langle v| H^\theta$ for $v \in \{0,1\}$. From the transcript and the trapdoor information, the verifier can compute v; in contrast, the prover, who does not know the trapdoor, cannot efficiently compute θ or v. In the last round, the verifier sends θ to the prover, who returns $v' \in \{0,1\}$; the verifier then checks whether $v' = v$. The honest prover would generate v' by measuring its remaining qubit $H^\theta |v\rangle\langle v| H^\theta$ in the basis θ and therefore always pass the verifier's check.

We can model this last round of the protocol (with a potentially dishonest prover) as follows: at the start, the prover has a state $\sigma^{(\theta,v)}$, which it produced as a result of the previous rounds of the protocol. For an honest prover, $\sigma^{(\theta,v)} = H^\theta |v\rangle\langle v| H^\theta$. Of course, this state can depend on all of $\texttt{ts}$, but we only make the dependence on θ and v explicit. Upon receiving $\theta \in \{0,1\}$ the prover measures a binary observable Z (if $\theta = 0$) or X (if $\theta = 1$) and returns the outcome v'. An honest prover would simply use the Pauli observables $Z = \sigma_Z$ and $X = \sigma_X$. The key step in the proof of [24] is to show that, due to the properties of ENTCF families, for any (potentially dishonest) prover that is accepted with high probability, the observables X and Z must anti-commute when acting on the prover's state. Then, Theorem 1 (for $n = 1$) follows from known results [34, 39, 26].

For our parallel RSP protocol we run n independent copies of the protocol from [24] in parallel, except that the basis choice θ_i is the same for each copy.[1] The prover's state before the last round of each copy of the RSP protocol is now denoted by $\sigma^{(\theta,\vec{v})}$, where $\vec{v} \in \{0,1\}^n$

[1] The advantage of this is that a prover that succeeds with high probability on average over θ must also succeed with high probability for each θ individually. If we were to sample θ independently for each of the parallel copies we could not conclude that a prover succeeds with high probability for any particular choice of $\theta_1, \ldots, \theta_n$ as there are exponentially many such choices.

can be calculated by the verifier from the transcript ts by repeating the same calculation as above for each parallel copy. Generalising from the single-qubit case, given $\theta \in \{0,1\}$ the prover performs a measurement to generate $\vec{v} \in \{0,1\}^n$, which we can describe by binary observables Z_i, X_i (for $\theta = 0, 1$ respectively) that correspond to the observable used to produce the i-th entry of $\vec{v}$. (For an honest prover, $\sigma^{(\theta,\vec{v})} = H^\theta |v_1\rangle\langle v_1| H^\theta \otimes \ldots \otimes H^\theta |v_n\rangle\langle v_n| H^\theta$ and Z_i is a Pauli-Z measurement on the i-th qubit.)

The main challenge in the proof is to establish that the prover must treat all of the parallel copies of the RSP protocol independently, i.e. to show that its (a priori uncharacterised) Hilbert space can be partitioned into n identical subspaces, one for each copy of the protocol. At first sight, it might look as though for this it suffices to show that X_i and Z_j (approximately) commute for all $i \neq j$. However, this is not the case because any such commutation statement can only be shown in a special *state-dependent distance* [47], which does not allow us to combine individual commutation statements into the *global* statement that the Hilbert space factorises into n subspaces. Instead, we need to consider the family $\{Z(\vec{a})X(\vec{b})\}_{\vec{a},\vec{b}\in\{0,1\}^n}$ of 4^n binary observables, where $Z(\vec{a}) = Z_1^{a_1} \cdots Z_n^{a_n}$. We then have to show that $\{Z(\vec{a})X(\vec{b})\}$ form an approximate representation of the Pauli group [26, 48].[2] This means that when acting on the prover's (unknown) state $\sigma^{(\theta)}$ (where $\sigma^{(\theta)}$ is like $\sigma^{(\theta,\vec{v})}$, but averaged over all $\vec{v}$), the operators $\{Z(\vec{a})X(\vec{b})\}$ behave essentially like Pauli operators. Formally, this means showing that on average over $\vec{a}, \vec{b} \in \{0,1\}^n$,

$$\mathrm{Tr}\Big[Z(\vec{a})X(\vec{b})Z(\vec{a})X(\vec{b})\sigma^{(\theta)}\Big] \approx (-1)^{\vec{a}\cdot\vec{b}}. \tag{2.1}$$

This is the appropriate generalisation of the statement that Z and X anti-commute in the single-qubit case. It is easy to check that Equation (2.1) holds when Z_i and X_i are the Pauli observables.

Our proof of Equation (2.1) has five main steps, which we briefly sketch here with references to the corresponding parts of the formal proof.

(1) Instead of working with the observables X_i, we define "inefficient observables" $\tilde{X}_i = (-1)^{v_i} X_i$, where v_i is the i-th bit of the verifier's string $\vec{v}$. $\tilde{X}_i$ is not an observable that an efficient prover can implement because it depends on v_i, which requires the trapdoor information to be computed efficiently. Intuitively, while X_i describes the prover's answer, $\tilde{X}_i$ describes whether that answer is accepted by the verifier. This has the advantage that the state $\sigma^{(\theta=1)}$ (averaged over $\vec{v}$) of a successful prover is an approximate $+1$-eigenstate of $\tilde{X}_i$, but not of X_i.

(2) We extend the family of states $\{\sigma^{(\theta)}\}_{\theta\in\{0,1\}}$ to a larger family of "counterfactual states" $\{\sigma^{(\vec{\theta})}\}_{\vec{\theta}\in\{0,1\}^n}$, which are defined as the states the prover would have prepared if the verifier had sent keys $k_i \in \mathcal{K}_{\theta_i}$. In Protocol 1 the basis choice is the same for all i, i.e. $\vec{\theta} = \vec{0}$ or $\theta = \vec{1}$, so for other choices of $\vec{\theta}$ these states are never actually prepared. However, they are still well-defined because for any prover in the actual protocol, we can fix that prover's operations (as a quantum circuit acting on a given input) and then consider what state those operations would produce if given keys with an arbitrary basis choice $\vec{\theta}$. The reason these counterfactual states are useful is that we can show that, as a consequence of the properties of ENTCF families, the states $\{\sigma^{(\vec{\theta})}\}_{\vec{\theta}}$ are computationally indistinguishable.

[2] When we say "Pauli group" we always mean the Pauli group modulo complex conjugation, which is also sometimes called the Heisenberg-Weyl group.

(3) We now want to show various commutation and anti-commutation relations for the observables $Z(\vec{a})$ and $\tilde{X}(\vec{b})$. For example, we want to show that Z_i and $\tilde{X}_i$ anti-commute, but Z_i and $\tilde{X}_j$ commute (for $i \neq j$). To show these relations, we make use of the counterfactual states $\sigma^{(\vec{\theta})}$ in the following way: for any particular relation, we can pick a $\vec{\theta}$ that makes showing this relation especially convenient. For example, to show that Z_i and $\tilde{X}_j$ commute, we would choose a $\vec{\theta}$ with $\theta_i = 0$ and $\theta_j = 1$ since the verifier can check the outcomes of "Z-type observables" for $\theta = 0$ and "X-type observables" for $\theta = 1$. Using the properties of ENTCF families, we can argue that the prover's measurements on these counterfactual states still yield outcomes that would pass the verifier's checks for each choice of θ_i. Based on this, we can show the desired relations for a "convenient" choice of counterfactual state $\sigma^{(\vec{\theta})}$. Then, we can relate these statements back to the prover's actual states $\sigma^{(\theta)}$ using the computational indistinguishability of $\{\sigma^{(\vec{\theta})}\}$. This is somewhat delicate because $\tilde{X}_i$ are inefficient.

(4) We can combine the various commutation and anti-commutation statements from the previous step to show that the observables $\{Z(\vec{a})\tilde{X}(\vec{b})\}$ behave like Pauli observables on $\sigma^{(\theta=1)}$, i.e. we show Equation (2.1) but with $\tilde{X}$ instead of X. This step relies on the fact that $\sigma^{(\theta=1)}$ is an approximate $+1$-eigenstate of $\tilde{X}(\vec{b})$ for all $\vec{b}$.

(5) Since we now know that $\{Z(\vec{a})\tilde{X}(\vec{b})\}$ behave essentially like Pauli observables, we can define an explicit isometry $\tilde{V}$ which can be shown to map $\{Z(\vec{a})\tilde{X}(\vec{b})\}$ to the corresponding Pauli observables. This means that we have good control over these inefficient observables, and we know how the inefficient and efficient observables are related. We can use this to define a modified isometry V that maps the efficient observables $\{Z(\vec{a})X(\vec{b})\}$ to the corresponding Pauli observables. This is a stronger version of Equation (2.1) and, combined again with the verifier's checks in the protocol and properties of ENTCF families, can be used to show that the prover must have prepared BB84 states.

We briefly comment on the relation between our soundness proof and that in [36]. At a high level, the soundness proof in [36] also shows a kind of "parallel rigidity" of two executions of a remote state preparation protocol. However, their proof proceeds quite differently from ours: they first show that observables "on the first qubit" anti-commute, which allows them to make a partial statement about the prover's state. This in turn can be used to extend the statement about the prover's observables to two-qubit observables, which is finally used to prove a statement about the prover's two-qubit state. This qubit-by-qubit approach is extremely costly in terms of parameters due to switching back and forth between making partial statements about the observables and state, and cannot reasonably be extended to n qubits. In contrast, we can make a global statement about the prover's 4^n possible observables without first characterising parts of the prover's state. This allows us to prove a parallel rigidity statement for n qubits without an exponential degradation of parameters.

3 Applications

Having introduced our parallel RSP theorem, we can turn to its cryptographic applications. We consider various cryptographic primitives that have previously been defined and constructed in a setting where one party sends random BB84 states to the other. For each primitive, we give a formal definition of the "classical-client version" and show that this definition can be satisfied using our parallel RSP protocol as a building block. Since our parallel RSP protocols relies on the LWE assumption, so do the dequantised protocols we present here. Furthermore, Theorem 1 only guarantees the preparation of BB84 states up to an inverse polynomial error, so as a result, the dequantised protocols only have inverse polynomial security (see Section 5 for a discussion of this point). Some of these primitives have previously been dequantised using an application-specific approach (and similarly relying on computational assumptions) [24, 13, 42, 28, 32]; in contrast, our approach is *generic* and simply uses RSP to replace the sending of BB84 states. We give a short overview of the different applications and refer to the full manuscript for details.

Unclonable quantum encryption. As a first application of our parallel RSP protocol, we consider the notion of *unclonable quantum encryption.* This cryptographic functionality was coined by Gottesman [25] and then formalised by Broadbent and Lord [11]. In a private-key unclonable quantum encryption scheme, a classical message is encrypted into a quantum state (the *quantum ciphertext*) with the following property: given only a single quantum ciphertext, it is impossible to create two states that can later both be decrypted with access to the private key. We consider an unclonable conjugate coding *hybrid encryption* scheme which is inspired by the work of Broadbent and Lord: a plaintext $\vec{m} \in \{0,1\}^n$ is encrypted with a randomly chosen secret key $k = (\vec{s}, \vec{\theta}) \xleftarrow{\$} \{0,1\}^n \times \{0,1\}^n$ and randomness $\vec{v} \xleftarrow{\$} \{0,1\}^n$ into the quantum ciphertext given by $\mathsf{Enc}_k(\vec{m}) = \bigotimes_{i=1}^n H^{\theta_i} |v_i\rangle\langle v_i| H^{\theta_i} \otimes |\vec{v} \oplus \vec{s} \oplus \vec{m}\rangle\langle \vec{v} \oplus \vec{s} \oplus \vec{m}|$. To decrypt using the secret key $k = (\vec{s}, \vec{\theta})$, one applies $H^{\theta_1} \otimes \cdots \otimes H^{\theta_n}$ to the first half of the ciphertext, measures in the computational basis with outcome $\vec{x}$, and then uncomputes the one-time pad in the second half using $\vec{x}$ and $\vec{s}$. The fact that this scheme is unclonable is a consequence of the *monogamy of entanglement* [11, 46].

To dequantise this protocol, we consider a scenario in which a classical client $\mathcal{C}$ wishes to delegate an unclonable ciphertext to a quantum receiver $\mathcal{R}$. As a first step, $\mathcal{C}$ and $\mathcal{R}$ run our parallel RSP protocol to delegate a collection of random BB84 states of the form $H^{\theta_1}|v_1\rangle \otimes \cdots \otimes H^{\theta_n}|v_n\rangle$, where $\vec{v}, \vec{\theta} \in \{0,1\}^n$ are random strings known only to $\mathcal{C}$. Then, $\mathcal{C}$ can choose $\vec{s} \in \{0,1\}^n$ and output the string $\vec{v} \oplus \vec{s} \oplus \vec{m}$ and set $\vec{k} = (\vec{s}, \vec{\theta})$ as the secret key. With this choice of key, the delegated parallel BB84 states are exactly the ciphertext $\mathsf{Enc}_k(\vec{m})$. Because the final output state of the protocol is computationally indistinguishable from a tensor product of BB84 states (known to the client), we can follow a similar proof as in [11] to obtain a classical-client unclonable encryption scheme with inverse-polynomial security.

Quantum copy-protection. In quantum copy-protection (QCP), a vendor wishes to encode a program into a quantum state in a way that enables a recipient to run the program, but not to create functionally equivalent "pirated" copies. The notion of QCP was introduced by Aaronson [1], who gave the first construction for unlearnable and efficiently computable functions in a strong quantum oracle model, which has since been improved to only requiring classical oracles [2]. Recent work [16] has also provided the first construction of QCP for compute-and-compare programs in the quantum random oracle model (QROM) as well as a scheme for multi-bit point functions in the QROM based on unclonable encryption with wrong-key detection (WKD) – a property which enables the decryption procedure to recognise incorrect keys.

Our QCP scheme for multi-bit point functions combines our unclonable hybrid encryption scheme with the generic WKD transformation in the QROM proposed by Coladangelo et al. [16]. The basic idea behind our QCP scheme is as follows. To encode a point function $P_{\vec{y},\vec{m}}$ (which is defined as returning $\vec{m}$ on input $\vec{y}$ and 0^n, otherwise) we simply output $\mathsf{Enc}_{\vec{y}}(\vec{m})$ together with $h(\vec{y})$, where h is a suitable hash function which we model as a truly random function (in the QROM). To evaluate the program on an input $\vec{x} \in \{0,1\}^{2n}$, we first check whether $\vec{x}$ hashes to $h(\vec{y})$ under h. If true, we decrypt as in the aforementioned hybrid encryption scheme and recover $\vec{m}$. Otherwise, we output 0^n.

We then show how to obtain a QCP scheme with a classical client through the use of our parallel RSP protocol for preparing random BB84 states, similar to our aforementioned classical-client unclonable encryption scheme. Our scheme enables a classical client to delegate a correct copy-protected program from the class of multi-bit point functions consisting of uniformly random marked inputs $\vec{y}$ and output strings $\vec{m}$ with inverse-polynomial security.

Quantum computing on encrypted data. Suppose a client wishes to perform some quantum computation, represented as the action of a quantum circuit C on an input state $|x\rangle$, with $x \in \{0,1\}^n$. For simplicity, we will assume the desired output is classical and corresponds to a computational basis measurement of $C|x\rangle$. The client only has limited quantum capability and therefore wishes to delegate the computation to a quantum server while ensuring the privacy of the input $|x\rangle$ and the output resulting from the measurement of $C|x\rangle$. Essentially, the client would like to send the server an encryption of the input and, after performing an interactive protocol, obtain an encryption of the output (which the client can decrypt, but the server cannot)[3]. This primitive is called quantum computing on encrypted data (QCED).

Many protocols for QCED with differing quantum requirements on the client have been developed (see [20] for a survey). Here we will focus on the protocol of Broadbent [8] which achieves QCED with a client that is only required to prepare BB84 states and send them to the server. This makes the protocol well-suited for dequantisation via our parallel RSP protocol. Before explaining this dequantisation, we (informally) define what a QCED protocol *with a classical client* should achieve. As before, the client's input is the string $x \in \{0,1\}^n$ and the goal is to obtain the outcome of measuring $C|x\rangle$ in the computational basis. In contrast to before, this must be achieved using only *classical* interaction with the quantum server. The requirement that the client's input must stay private is captured by the condition that after interacting with the client, it must be computationally intractable for the server to decide which one of two distinct inputs the client used.

Our QCED protocol with a classical client works as follows. The client first performs the parallel RSP protocol with the server, resulting in the preparation of BB84 states (or the client aborting). Provided the protocol succeeded, the client proceeds to run Broadbent's protocol [8] as if the server had received those BB84 states via a quantum channel. The security proof is straightforward. First, we know that after performing RSP the server's state is computationally indistinguishable from a tensor product of BB84 states (known to the client). Furthermore, the interaction in [8] preserves this computational indistinguishability. Hence, the server's state at the end of the protocol is indistinguishable from the state the server would have obtained by executing the protocol with random BB84 states and the security of our protocol follows from [8].

Verifiable delegated blind quantum computation. The final application we consider is verifiable delegated blind quantum computation (VDBQC). VDBQC is an interactive protocol between two parties, in this case denoted as the verifier and the prover. The verifier delegates a computation to the prover and, in addition to ensuring input-output privacy as in QCED, the protocol also ensures that the probability for the verifier to accept an incorrect output is small. In other words, if the prover deviates from the protocol and does not perform the verifier's instructed computation, the verifier should be able to detect this and abort with high probability. As with QCED, a number of such protocols have been developed and we refer the reader to [23] for a survey.

[3] This also allows the client to hide the computation itself from the server by suitably encoding it as part of the input x and taking C to be a *universal circuit*. When the primary goal of the protocol is to hide the computation, it is referred to as a *blind quantum computing protocol* [3, 9].

Here we focus on a protocol by Morimae [38]. This protocol achieves verifiability by combining a protocol for blind quantum computation (or QCED) with the *history state construction*, which is a special encoding of a quantum circuit into a quantum state [31, 30]. In Morimae's protocol, for a given circuit C the verifier uses a QCED protocol to delegate to the prover the preparation of two such history states (one for C and one for the complement of C, where the output qubit is negated). The verifier then requests these states from the prover and proceeds to measure them in the computational or Hadamard basis. This allows the verifier to determine the output of the computation. The history state construction guarantees that malicious behavior on the prover's part would be detected by the verifier's measurement. Additionally, the use of a QCED protocol ensures that the prover is "blind", i.e. does not know which computation the verifier delegated.

To dequantise this protocol, we use our QCED protocol with a classical client to delegate the preparation of the two history states to the quantum prover. We then replace the verifier's measurements on this state by a *measurement protocol* due to Mahadev [33], which allows the classical verifier to delegate these measurements to the prover in a way that forces the prover to report the correct outcomes. We thus obtain a VDBQC protocol with a classical verifier. Crucially, through the use of the classical client QCED protocol and Mahadev's measurement protocol, the prover is "computationally blind", i.e. unable to distinguish which computation the verifier has performed. In contrast, Mahadev's verification protocol [33] does not have this property.[4]

4 Related work

A number recent of works starting with [7, 33] have developed techniques that allow a classical verifier to use post-quantum cryptography to force an untrusted (but computationally bounded) quantum prover to behave in a certain way. Here, we briefly describe these works and explain their relation to our parallel RSP protocol.

In a breakthrough result [33], Mahadev introduced a protocol that allows a classical verifier to delegate a quantum computation to a quantum computer and be able to verify the correctness of the result. The key ingredient for this protocol is a *measurement protocol*, which allows the verifier to securely delegate single-qubit measurements in the standard or Hadamard basis to a quantum prover, assuming that the prover cannot break the LWE assumption. This can then be applied to so-called *prepare-and-measure protocols*: if one has a protocol that involves a quantum prover preparing and sending a quantum state to the verifier and the verifier performing single-qubit measurements on this state, one can use Mahadev's measurement protocol to delegate these quantum measurements to the prover itself. This yields a protocol in which the prover only sends classical measurement outcomes to the verifier, hence making the verifier classical.

This measurement protocol is in many ways similar to what we seek to do in this paper: it removes the need for quantum communication between a fully quantum prover and a verifier with very limited quantum capabilities (only measuring single qubits in the computational or Hadamard basis). The difference to our work is that we are concerned with *prepare-and-send protocols*, in which the verifier sends random BB84 states to the prover instead of receiving them.

[4] In [24], the authors also construct a blind verification protocol based on RSP. However, they approach the problem in a composable framework, which requires them to make an additional assumption on the prover (called the *measurement buffer* in [24]). In contrast, our protocol requires no extra assumptions on the prover. We describe the issue with the measurement buffer assumption in more detail in Section 4.

It turns out that replacing the quantum communication of prepare-and-send protocols requires significantly stronger control over the untrusted prover. At a high level, the reason is the following: for Mahadev's measurement protocol, it suffices to show that *there exists* a quantum state that is consistent with the distribution of measurement outcomes reported by the prover, in the sense that the measurement outcomes for different bases could have been obtained by measurements on (copies of) the same state. In contrast, if we want to replace the step of the verifier sending a physical quantum state to the prover, we need to show that the prover has *actually constructed* a certain quantum state, not just that such a quantum state exists mathematically.[5] We give a more detailed description of what it means to "actually construct" a quantum state in Section 2.1.

The first classical protocol that provably forced a quantum prover to prepare a certain quantum state was the single-qubit RSP protocol of [24] (see also [13] for a related result). This protocol essentially achieves our informal theorem as stated above for a single qubit, i.e. $n = 1$.[6] At first sight, it might seem as though a simple hybrid argument, which replaces each BB84 qubit with a (sequential) instance of [24], suffices to achieve the multi-qubit task. However, the single-qubit RSP protocol of [24] only ensures that each BB84 qubit can be individually replaced by an RSP protocol up to a *global* isometry. Because the prover's state can be entangled in arbitrary ways between intermediate applications of the protocol, it is difficult to justify that all of the individual replacements together form an actual n-qubit BB84 state; as we explain below, the fact that the protocol from [24] is composable does not remedy this situation, either. While some prior work [19] showed that composable *single-qubit* RSP suffices in the context of quantum verification, one would have to show a similar result for each application of interest. Our parallel RSP protocol, in contrast, can be used in a plug-and-play manner for many cryptographic protocols and applications. In addition, our protocol has fewer rounds than a sequential repetition of [24] and also immediately yields a *proof of quantum space* (a certificate that the prover has a certain number of qubits). We give a brief outline of [24] and its soundness proof in Section 2.1.

The main difficulty in going from [24] to our parallel RSP result is enforcing a tensor product structure on the prover's space: we would like to show that, if we execute multiple instances of a single-qubit RSP protocol in parallel, a successful prover must treat each of these copies independently. Mathematically, this means that we need to be able to split the prover's a priori uncharacterised Hilbert space into a *tensor product*, where each tensor factor is supposed to correspond to one instance of the RSP protocol. This is a more demanding version of the classic question of parallel repetition: there, one is interested in showing that any prover's winning probability in the protocol decays in essentially the same way as it would for a prover who executes the instances independently. In contrast, we need to show that the prover really does execute the different instances independently in a physically meaningful sense. We call this stronger requirement *parallel rigidity.*

In [24], the authors show that their protocol has composable security. This may suggest that one can obtain a parallel rigidity statement simply by composing the protocol with itself in sequence or in parallel. However, this is not the case because the composable security

[5] In fact, in [49] it was shown that Mahadev's measurement protocol does ensure that the prover *knows* (in the sense of a proof of knowledge) the state it is measuring, not just that it exists mathematically. The notion of "knowing" a quantum state is quite subtle to define and we forego a detailed description here, but point out that this is weaker than showing that the prover actually constructed the state and (to the best of our knowledge) not sufficient to use Mahadev's protocol for prepare-and-measure scenarios.

[6] The protocol in [24] allows for the qubit to be prepared in one of 10 possible states which includes the 4 BB84 states. Here, we only focus on the 4 BB84 states as this is the case we will deal with in our parallel RSP protocol.

statement in [24] requires an additional assumption called a *measurement buffer*, which effectively acts as a trusted intermediary between the verifier and the prover. A sequential or parallel composition of the protocol in [24] would utilise a different measurement buffer for each instance, thereby forcing the prover to treat the different instances in a (largely) independent way. In particular, this means that one already assumes a tensor product structure with n separate qubits in the prover's space, whereas in our work enforcing this tensor product structure is the key technical challenge. For cryptographic applications, we do not want to place any such assumption on the prover and instead allow the prover to perform arbitrary global operations involving all instances. This is what our parallel RSP protocol achieves. Furthermore, as shown in [4], achieving a composable single-qubit RSP without the measurement buffer is impossible. This means that one cannot hope to achieve parallel RSP by showing a stronger composable version of single-qubit RSP; instead, it is necessary to directly analyse parallel executions of the protocol, as we do in this paper.

The question of parallel rigidity has been studied extensively in the literature on quantum self-testing [17, 14, 39, 40], where one considers a setting of two non-communicating provers. Unfortunately, those techniques are not immediately transferable to the setting we consider here, namely a single computationally bounded prover.

Some progress towards the question of parallel rigidity for single computationally bounded provers was made in [36], which gives a protocol that allows a classical verifier to certify that a quantum prover must have prepared and measured a Bell state, i.e. an entangled 2-qubit quantum state. This has since been applied to device-independent quantum key distribution [35] and oblivious transfer [12], and been extended to work for magic states [37]. The protocol from [36] uses a 2-fold parallel repetition of [24] (with additional steps to allow for the certification of an entangled state, not just product states). As part of their soundness proof, [36] do show a kind of parallel rigidity result for 2 instances of the RSP protocol. However, their method does not generalise to an n-fold parallel repetition without an exponential decay in parameters. Hence, for our n-fold parallel rigidity proof, new techniques are needed. A more detailed comparison between our new parallel rigidity proof and the method in [36] can be found at the end of Section 2.1. We note that in independent concurrent work, [21] also gave an n-fold parallel rigidity proof in the computational setting, but the class of states they deal with is different from random BB84 pairs and they do not consider the dequantisation of cryptographic protocols.

In addition to this line of work focused on rigidity statements, application-specific dequantisations were already considered for private-key quantum money [22, 42], certifiable deletion of quantum encryption [28] and secure software leasing [32]. In all these cases the authors derived the desired security statement from properties of trapdoor claw-free functions, a cryptographic primitive which is also the basis of our RSP protocol. While this is less generic and modular than our approach and requires a new analysis for each application, it does have the advantage that one can obtain *negligible* security, whereas with RSP we obtain inverse polynomial security. We comment more on the possibility of negligible security from RSP-like primitives in Section 5.

5 Discussion

We have shown how a classical verifier can certify a tensor product of BB84 states in the memory of a quantum prover, assuming the quantum-intractability of the LWE problem. Importantly, the prover does not know which BB84 states it has prepared, whereas the verifier does. Hence, the result at the end of the protocol is as if the verifier had sent random

BB84 states to the prover. This allows us to dequantise a number of quantum cryptographic primitives, yielding a generic and modular way of translating these protocols to a setting where only classical communication is used. We have demonstrated the versatility of this approach by applying it to unclonable encryption, quantum copy-protection, computing on encrypted data, and blind verification. Naturally, we expect that other primitives that rely on BB84 states can also be dequantised using our approach. Examples of this include quantum encryption with certified deletion [10, 41] and private key quantum money [50, 42]. We leave these and other applications to future work.

Apart from applying our technique to dequantise additional cryptographic primitives, our work raises a number of further open problems. Firstly, while our RSP primitive is based on the hardness of LWE, we can ask whether it is possible to achieve this functionality from weaker computational assumptions. For instance, would it be possible to perform an RSP-like protocol assuming only the existence of quantum-secure one-way functions? This is of particular interest because recent results have shown that secure two-party computation can be achieved from one-way functions and quantum communication [5, 27]. These results are based on the fact that an oblivious-transfer protocol can be implemented from one-way functions and quantum communication that consists of BB84 states. However, an RSP primitive like ours would allow one to generically dequantise that quantum communication. Hence if RSP (with sufficiently strong parameters) can be obtained from quantum-secure one-way functions, then secure two-party computation can also be obtained from those functions, together with classical communication. In light of earlier work [29, 45] we conjecture that this is impossible. Formalising this intuition could lead to a better understanding of the minimum assumptions required for performing RSP-like protocols.

Secondly, a more technical open problem concerns the parameters of our rigidity theorem, Theorem 1. As stated above, provided the prover accepts, the state the verifier certifies is $1/\operatorname{poly}(n)$-close to a tensor product of n BB84 states (up to an isometry). The $1/\operatorname{poly}(n)$ closeness means that the soundness error of our dequantised protocols also scales as $1/\operatorname{poly}(n)$. It would be desirable to achieve *negligible* soundness error, particularly when considering composable instances of these protocols. This is not possible with the approach taken in this paper as the statistical argument used in deriving our main theorem will necessarily introduce $1/\operatorname{poly}(n)$ factors. However, it might be possible to circumvent an explicit RSP statement: the advantage of the RSP statement in our paper is that one can use it to dequantise existing protocols easily, but these existing protocols typically only use BB84 states because of their no-cloning properties. Therefore, instead of using an RSP protocol to prepare those states, one could instead try to show a "post-quantum cryptographic no-cloning property" directly that could plausibly be used to dequantise these protocols while preserving negligible soundness.

Finally, we mention that our derivation of the parameters in the rigidity theorem is likely not optimal and could be optimised to improve the efficiency of our protocol. The situation here is similar to that of parallel self-testing in the multi-prover setting, with the first works having round complexity that scaled as a high-degree polynomial [44] and more recent works achieving quasilinear scaling [39, 15]. It would be interesting to see whether ideas from these newer works are also applicable in the setting of parallel remote state preparation.

References

1 Scott Aaronson. Quantum copy-protection and quantum money. *2009 24th Annual IEEE Conference on Computational Complexity*, July 2009. doi:10.1109/ccc.2009.42.

2 Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. New approaches for quantum copy-protection. In *Annual International Cryptology Conference*, pages 526–555. Springer, 2021.

3 Pablo Arrighi and Louis Salvail. Blind quantum computation. *International Journal of Quantum Information*, 4(05):883–898, 2006.

4 Christian Badertscher, Alexandru Cojocaru, Léo Colisson, Elham Kashefi, Dominik Leichtle, Atul Mantri, and Petros Wallden. Security limitations of classical-client delegated quantum computing. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 667–696. Springer, 2020.

5 James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. One-way functions imply secure computation in a quantum world. In *Annual International Cryptology Conference*, pages 467–496. Springer, 2021.

6 C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, pages 8, vol. 175, 1984.

7 Z. Brakerski, P. Christiano, U. Mahadev, U. Vazirani, and T. Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. *IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 320–331, 2018. doi:10.1109/FOCS.2018.00038.

8 Anne Broadbent. Delegating private quantum computations. *Canadian Journal of Physics*, 93(9):941–946, 2015.

9 Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 517–526. IEEE, 2009.

10 Anne Broadbent and Rabib Islam. Quantum encryption with certified deletion. In *Theory of Cryptography*, pages 92–122. Springer International Publishing, 2020. doi:10.1007/978-3-030-64381-2_4.

11 Anne Broadbent and Sébastien Lord. Uncloneable Quantum Encryption via Oracles. In Steven T. Flammia, editor, *15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020)*, volume 158 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:22, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.TQC.2020.4.

12 Anne Broadbent and Peter Yuen. Device-independent oblivious transfer from the bounded-quantum-storage-model and computational assumptions. *arXiv preprint*, 2021. arXiv:2111.08595.

13 Alexandru Cojocaru, Léo Colisson, Elham Kashefi, and Petros Wallden. QFactory: Classically-instructed remote secret qubits preparation. *Advances in Cryptology - ASIACRYPT 2019, Lecture Notes in Computer Science, Springer*, pages 615–645, 2019. doi:10.1007/978-3-030-34578-5_22.

14 Andrea Coladangelo. Parallel self-testing of (tilted) EPR pairs via copies of (tilted) CHSH. *arXiv preprint*, 2016. arXiv:1609.03687.

15 Andrea Coladangelo, Alex B Grilo, Stacey Jeffery, and Thomas Vidick. Verifier-on-a-leash: new schemes for verifiable delegated quantum computation, with quasilinear resources. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 247–277. Springer, 2019.

16 Andrea Coladangelo, Christian Majenz, and Alexander Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model. *arXiv preprint*, 2020. arXiv:2009.13865.

17 Matthew Coudron and Anand Natarajan. The parallel-repeated magic square game is rigid. *arXiv preprint*, 2016. arXiv:1609.06306.

18 W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976. doi:10.1109/TIT.1976.1055638.

19 Vedran Dunjko and Elham Kashefi. Blind quantum computing with two almost identical states, 2016. doi:10.48550/arXiv.1604.01586.

20 Joseph F Fitzsimons. Private quantum computation: an introduction to blind quantum computing and related protocols. *npj Quantum Information*, 3(1):1–11, 2017.

21 Honghao Fu, Daochen Wang, and Qi Zhao. Computational self-testing of multi-qubit states and measurements. *arXiv preprint*, 2022. `arXiv:2201.13430`.

22 Dmitry Gavinsky. Quantum money with classical verification. In *2012 IEEE 27th Conference on Computational Complexity*, pages 42–52. IEEE, 2012.

23 Alexandru Gheorghiu, Theodoros Kapourniotis, and Elham Kashefi. Verification of quantum computation: An overview of existing approaches. *Theory of computing systems*, 63(4):715–808, 2019.

24 Alexandru Gheorghiu and Thomas Vidick. Computationally-secure and composable remote state preparation. *IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1024–1033, 2019. `doi:10.1109/FOCS.2019.00066`.

25 Daniel Gottesman. Uncloneable encryption. *Quantum Information and Computation*, pages 3:581–602, 2003.

26 William Timothy Gowers and Omid Hatami. Inverse and stability theorems for approximate representations of finite groups. *Sbornik: Mathematics*, 208(12):1784, 2017. `doi:10.1070/SM8872`.

27 Alex B Grilo, Huijia Lin, Fang Song, and Vinod Vaikuntanathan. Oblivious transfer is in miniqcrypt. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 531–561. Springer, 2021.

28 Taiga Hiroka, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa. Quantum encryption with certified deletion, revisited: Public key, attribute-based, and classical communication. In *Advances in Cryptology – ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part I*, pages 606–636, Berlin, Heidelberg, 2021. Springer-Verlag. `doi:10.1007/978-3-030-92062-3_21`.

29 Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 44–61, 1989.

30 Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. *Siam journal on computing*, 35(5):1070–1097, 2006.

31 Alexei Yu Kitaev, Alexander Shen, Mikhail N Vyalyi, and Mikhail N Vyalyi. *Classical and quantum computation*, volume 47. American Mathematical Soc., 2002.

32 Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Secure software leasing from standard assumptions. In *Theory of Cryptography Conference*, pages 31–61. Springer, 2021.

33 Urmila Mahadev. Classical verification of quantum computations. *IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–267, 2018. `doi:10.1109/FOCS.2018.00033`.

34 M McKague, T H Yang, and V Scarani. Robust self-testing of the singlet. *Journal of Physics A: Mathematical and Theoretical*, 45(45):455304, October 2012. `doi:10.1088/1751-8113/45/45/455304`.

35 Tony Metger, Yfke Dulek, Andrea Coladangelo, and Rotem Arnon-Friedman. Device-independent quantum key distribution from computational assumptions. *New Journal of Physics*, 23(12):123021, 2021.

36 Tony Metger and Thomas Vidick. Self-testing of a single quantum device under computational assumptions. *arXiv preprint*, 2020. `arXiv:2001.09161`.

37 Akihiro Mizutani, Yuki Takeuchi, Ryo Hiromasa, Yusuke Aikawa, and Seiichiro Tani. Computational self-testing for entangled magic states. *arXiv preprint*, 2021. `arXiv:2111.02700`.

38 Tomoyuki Morimae. Blind quantum computing can always be made verifiable. *arXiv preprint*, 2018. `arXiv:1803.06624`.

39 Anand Natarajan and Thomas Vidick. A quantum linearity test for robustly verifying entanglement. *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1003–1015, 2017. doi:10.1145/3055399.3055468.

40 Anand Natarajan and Thomas Vidick. Low-degree testing for quantum states, and a quantum entangled games PCP for QMA. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 731–742. IEEE, 2018.

41 Alexander Poremba. Quantum proofs of deletion for learning with errors, 2022. doi:10.48550/ARXIV.2203.01610.

42 Roy Radian and Or Sattath. Semi-quantum money. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, AFT '19, pages 132–146, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3318041.3355462.

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

44 Ben W Reichardt, Falk Unger, and Umesh Vazirani. Classical command of quantum systems. *Nature*, 496(7446):456–460, 2013.

45 Steven Rudich. The use of interaction in public cryptosystems. In *Annual International Cryptology Conference*, pages 242–251. Springer, 1991.

46 Marco Tomamichel, Serge Fehr, Jedrzej Kaniewski, and Stephanie Wehner. A monogamy-of-entanglement game with applications to device-independent quantum cryptography. *New Journal of Physics*, 15(10):103002, October 2013. doi:10.1088/1367-2630/15/10/103002.

47 Thomas Vidick. *The complexity of entangled games.* PhD thesis, UC Berkeley, 2011. URL: https://digitalassets.lib.berkeley.edu/etd/ucb/text/Vidick_berkeley_0028E_11907.pdf.

48 Thomas Vidick. Course FSMP, Fall'20: Interactions with quantum devices. http://users.cms.caltech.edu/~vidick/teaching/fsmp/fsmp.pdf, 2020.

49 Thomas Vidick and Tina Zhang. Classical proofs of quantum knowledge. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 630–660. Springer, 2021.

50 Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, January 1983. doi:10.1145/1008908.1008920.

51 William K Wootters and Wojciech H Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.