A computational test of quantum contextuality, and even simpler proofs of quantumness

Atul Singh Arora[*], Kishor Bharti[†], Alexandru Cojocaru[‡], Andrea Coladangelo[§]

Abstract

Bell non-locality is a fundamental feature of quantum mechanics whereby measurements performed on "spatially separated" quantum systems can exhibit correlations that cannot be understood as revealing predetermined values. This is a special case of the more general phenomenon of "quantum contextuality", which says that such correlations can occur even when the measurements are not necessarily on separate quantum systems, but are merely "compatible" (i.e. commuting). Crucially, while any non-local game yields an experiment that demonstrates quantum advantage by leveraging the "spatial separation" of two or more devices (and in fact several such demonstrations have been conducted successfully in recent years), the same is not true for quantum contextuality: finding the contextuality analogue of such an experiment is arguably one of the central open questions in the foundations of quantum mechanics.

In this work, we show that an arbitrary contextuality game can be compiled into an operational "test of contextuality" involving a single quantum device, by only making the assumption that the device is computationally bounded. Our work is inspired by the recent work of Kalai et al. (STOC '23) that converts any non-local game into a classical test of quantum advantage with a single device. The central idea in their work is to use cryptography to enforce spatial separation within subsystems of a single quantum device. Our work can be seen as using cryptography to enforce "temporal separation", i.e. to restrict communication between sequential measurements.

Beyond contextuality, we employ our ideas to design a "proof of quantumness" that, to the best of our knowledge, is arguably even simpler than the ones proposed in the literature so far.

[*]Joint Center for Quantum Information and Computer Science (QuICS), University of Maryland. IQIM, Caltech.

[†]A*STAR Quantum Innovation Centre (Q.InC), Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), Singapore. Centre for Quantum Engineering, Research and Education, TCG CREST, India.

[‡]School of Informatics, University of Edinburgh, UK. QuICS, University of Maryland.

[§]Paul G. Allen School of Computer Science and Engineering, University of Washington, USA.

Contents

1 Introduction

One of the most intriguing features of quantum mechanics is that, in general, observable properties of a quantum system, usually referred to as "observables", do not seem to hold a precise value until they are measured. In technical jargon, quantum mechanics is not a "local hidden variable theory". While this feature is now well-understood, Einstein, Podolski, and Rosen, in their paper [EPR35], originally conjectured that a local hidden variable explanation of quantum mechanics should exist. It was only years later that Bell [Bel64], and subsequently Clauser, Horne, Shimony, and Holt (CHSH) [CHSH69], in their seminal works, proposed an operational test, i.e. an experiment, capable of ruling out a local hidden variable explanation of quantum mechanics. More precisely, they showed that there exists an experiment involving measurements on "spatially separated" quantum systems such that the outcomes exhibit correlations that cannot be explained by a local hidden variable theory. Such an experiment, usually referred to as a Bell test or a *non-local game*, has been performed convincingly multiple times [HBD+15, GVW+15, SMSC+15, LWZ+18, RBG+17, SSK+23]. Crucially, a Bell test rules out a local hidden variable theory under the assumption that the devices involved in the experiment are non-communicating (which is usually enforced through "spatial separation").

Bell non-locality may be viewed as a special case of the more general phenomenon of quantum contextuality [Spe60, KS67], which says that such correlations can occur even when the measurements are not necessarily on separate quantum systems, but are merely "compatible" (i.e. commuting). Contextuality has a long tradition in the foundations of quantum mechanics [BCG+22]. However, unlike a non-local game, a more general *contextuality game*[1] does not in general have a corresponding operational test. By an operational test, we mean a test that can be carried out on a device by interacting with it classically, and importantly, without having to make bespoke assumptions about its inner workings. Thus, even though some contextuality games are even simpler than non-local games [KCBS08], a satisfactory approach to compiling arbitrary contextuality games into operational "tests of contextuality" is missing. This is not for lack of trying—numerous attempts have been made that inevitably have to either resort to strong assumptions about the quantum hardware or assumptions that are hard to enforce in practice, such as the device being essentially *memoryless*.[2] Thus, one of the central open questions in the foundations of quantum mechanics is:

Is there a way to compile an arbitrary contextuality game into an "operational test of contextuality"?

Beyond demonstrating the presence of genuine quantum behaviour, non-local games can be employed to achieve a much more fine-grained control over the behaviour of untrusted quantum devices: for example, they allow a classical user to verify the correctness of full-fledged quantum computations, by interacting with two non-communicating quantum devices [RUV13, CGJV19]. Of course, any guarantee obtained via non-local games hinges on the physical (and non-falsifiable) assumption that the devices involved are non-communicating. To circumvent the need for this assumption, a lot of the attention in recent years has shifted to the computational setting. This exploration was kick-started by Mahadev's seminal work [Mah18] showing, via cryptographic techniques, that the verification of quantum computations can be achieved with a single quantum device, under the assumption that the device is computationally bounded. She and her collaborators [BCM+21] later proposed what can be thought of as the analogue of a Bell/CHSH experiment with a single device—they proposed a simple test that an efficient quantum device can pass, but that an efficient classical device cannot. Since then, various works have proposed increasingly efficient "proofs of quantumness" in this setting [BKVV20, ACGH20, KMCVY22, KLVY23, AMMW22, BGKM+23]. The goal of this line of work is to simplify these tests to the point that they can be implemented on current quantum devices. An experimental realisation of such a proof of quantumness would be a milestone for the field of quantum computation, as it would constitute the first *efficiently verifiable* demonstration of quantum advantage. Towards this goal, the second question that we consider in this work is:

Can contextuality help realise simpler proofs of quantumness?

[1] We choose to use the term contextuality "game" here to preserve the analogy with a non-local game. However, in the contextuality literature, the more commonly used term is contextuality "scenario". We refer the reader to the technical overview (Section 2.2) for details.

[2] These assumptions are typically referred to as "loopholes" in the literature: [LLS+11, UZZ+13, JRO+16, ZKK+17, MZL+18, LMZ+18, ZXX+19, UZZ+20, WZL+22, HXA+23, LMX+23].

1.1 Our results

Our first contribution is a positive answer to the first question. We show that, using cryptographic techniques, an arbitrary contextuality game can be compiled into an "operational test of contextuality" involving a single quantum device, where the only assumption is that the device is computationally bounded.[3] A *contextuality game* involves a single player and a referee (unlike non-local games that always involve more than one player). In an execution of the game, the referee asks the player to measure a *context*—a set of commuting observables—and the player wins if the measurement outcomes satisfy certain constraints. Importantly, a given observable may appear in multiple contexts. If the player uses a strategy where the values of these observables are "predetermined" (which is the analogue of a "local hidden variable" strategy in the non-local game setting), then the highest winning probability achievable is referred to as the *non-contextual value* of the game, denoted by valNC.[4] On the other hand, if the player uses a quantum strategy, then the highest winning probability achievable is the *quantum value*, denoted by valQu (which is strictly greater than valNC for contextuality games of interest).[5] We formalise what we mean by an "operational test of contextuality" by introducing a property called *faithfulness*. Intuitively, this requires that the test has a strong correspondence to some contextuality game. We defer more precise definitions to Section 7. We show the following.

Theorem 1 (informal, simplified)**.** *Any contextuality game G can be compiled into a single prover, 2-round (i.e. 4-message) operational test of contextuality, under standard cryptographic assumptions. More precisely, the test is faithful to G and satisfies the following:*

- *(Completeness) There is a quantum polynomial-time (QPT) prover that wins with probability at least*

$$\frac{1}{2}(1 + \mathrm{valQu}) - \mathsf{negl}\,.$$

- *(Soundness) Any probabilistic polynomial-time (PPT) prover wins with probability at most*

$$\frac{1}{2}(1 + \mathrm{valNC}) + \mathsf{negl}\,.$$

Here negl are (possibly different) negligible functions of a security parameter.

The bounds in the statement above are for games with contexts of size two (which subsume "2-player" non-local games, see Figure 1). The general bounds are similar but depend on the size of the contexts and are stated later. Notably, the number of messages, even in the general case (which subsumes non-local games with any number of players), remains constant (i.e. four). The cryptographic assumptions we make are the existence of (1) a quantum homomorphic encryption (QFHE) scheme[6] and (2) a new primitive, the *oblivious pad*, which we introduce below. QFHE can be realised assuming the quantum hardness of the Learning With Errors problem (LWE) [Reg09]. We show how to construct the oblivious pad under the same assumption in the quantum random oracle model—which in turn can be heuristically instantiated using a cryptographic hash function, such as SHA3.[7]

Our result is motivated by the recent work of Kalai et al. [KLVY23] (KLVY from here on) that converts any non-local game into a test of quantumness with a single device. Consider a non-local game with two players, Alice and Bob. The central idea behind the KLVY compiler is to use cryptography to enforce spatial separation within a *single* quantum device, i.e. to enforce that Alice and Bob's measurements occur on separate subsystems. The KLVY compiler relies on the following mechanism to cryptographically enforce spatial separation: Alice's question and answer are encrypted (using a quantum fully homomorphic encryption (QFHE) scheme), while Bob's question and

[3]In the contextuality jargon, one might say that all "loopholes" are being replaced by a computational assumption.

[4]We emphasise that a non-contextual strategy is such that, if an "observable" appears in multiple contexts, then the *same* predetermined value should be returned by the player for that observable in *all* contexts in which it appears. This is precisely the difficulty in realising an "operational test of contextuality": how does the referee enforce that the player is consistent across contexts?

[5]A quantum strategy is such that, if an observable appears in multiple contexts, then the *same* observable should be measured to obtain the answer in all such contexts.

[6]With an assumption on the form of encrypted ciphertexts, which is satisfied by both Mahadev's and Brakerski's QFHE schemes, [Bra18, Mah20]; see Section 2.3.

[7]In the random oracle model [BR93] (ROM), a hash function f is modelled as a uniformly random black-box function: parties can evaluate it by sending a query x and receiving $f(x)$ in return. In the *quantum* random oracle model (QROM), such queries can also be made in superposition. A proof of security in this model is taken to be evidence for security of the protocol when the black-box is replaced by, for example, a suitable hash function f. This is because, informally, any attack on the resulting protocol must necessarily exploit the structure of f.

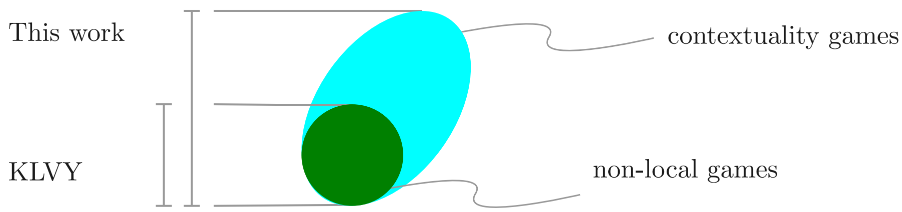


Figure 1: The compiler in this work compiles a much larger set of games compared to the one in [KLVY23].

answer are in the clear. The referee, who holds the decryption key, can then test the correlation across the two. Unfortunately, this approach does not extend to contextuality, at least not in any direct way, since in a contextuality game there is no notion of Alice and Bob.

In contrast, our work can be seen as using cryptography to enforce "temporal separation", i.e. to restrict communication between sequential measurements. In a nutshell, our idea is to ask the first question under a homomorphic encryption and the second question in the clear, as in KLVY, but with the following important difference. In KLVY, the quantum device prepares an *entangled* state, whose first half is encrypted, and used to homomorphically answer the first question, while the second half is not, and is used to answer the second question in the clear. In our protocol, there are no separate subsystems between the two rounds: instead, the encrypted post-measurement state from the first round (which results from the measurement performed to obtain the encrypted answer) is *re-used* in the second round. The technical barrier is, of course, the following: how can the post-measurement state be re-used if it is still encrypted? The two most natural approaches do not work:

(i) Providing the decryption key to the quantum device is clearly insecure as it allows the device to learn the first question in the clear.

(ii) Homomorphically encrypting the second question does not work either because the quantum device can correlate its answers to the two questions "under the hood of the homomorphic encryption".

We circumvent this barrier by introducing a procedure that allows the prover to *obliviously* "re-encrypt" the post-measurement state non-interactively. This re-encryption procedure is such that it allows the verifier to achieve the following: the verifier can now reveal some information that allows a quantum prover to access the post-measurement state in the clear, while a PPT prover *does not learn the first question.* More precisely, the verifier does not directly reveal the original decryption key, which would clearly be insecure, as pointed out earlier. Instead, the verifier expects the prover to "re-encrypt" its state using the new procedure, and then the verifier is able to safely reveal the resulting "overall" decryption key. Crucially, while a classical prover is unable to make use of the additional information to beat the classical value valNC, we show that there is an efficient quantum prover that can access the post-measurement state in the clear, and proceed to achieve the quantum value valQu.

The main technical tool that we introduce to formalise this idea, which may find applications elsewhere, is a primitive that we call *oblivious (Pauli) pad*. The oblivious pad takes as input a state $|\psi\rangle$ and a public key pk and produces a Pauli-padded state $X^x Z^z |\psi\rangle$ together with a string s (which can be thought of as encrypting x and z using pk). The string s can be used to recover x, z given the corresponding secret key sk. The security requirement is modelled as the following distinguishing game between a PPT prover and a challenger:

- The challenger generates public and secret keys $(\mathsf{pk}, \mathsf{sk})$ and sends pk to the prover.
- The prover produces a string s and sends it to the challenger.
- The challenger either returns (x, z) (as decoded using s and sk), or a fresh pad $(\tilde{x}, \tilde{z})$ sampled uniformly at random.

We require that no PPT prover can distinguish between the two cases with non-negligible advantage.[8] We show that an oblivious pad can be realised based on ideas from [BCM+21] in the quantum random oracle model. Crucially, while in KLVY the second phase of the protocol happens on a *different* subsystem, the oblivious pad is what allows us to carry out the second phase on the *same* subsystem, as depicted in Figure 2.

[8]Note that there is a QPT prover that *can* distinguish these two cases with non-negligible advantage.

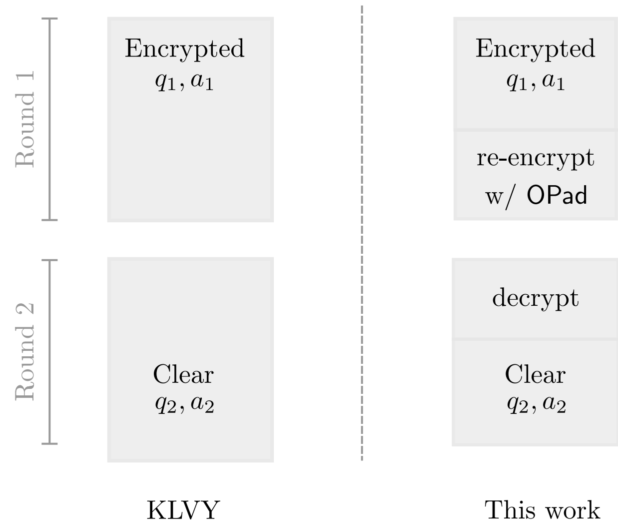


Figure 2: A schematic comparing the non-local game compiler [KLVY23] with our contextuality game compiler. The key idea in KLVY is to ask the first question of a non-local game under a homomorphic encryption and the second one in the clear, with the prover using two entangled subsystems (one that is encrypted, and one in the clear). In our compiler, the oblivious pad (OPad) allows the prover to "re-encrypt" its post-measurement state, just before Round 2. Upon obtaining information about the "re-encryption" that took place, the verifier can then safely reveal the "overall" decryption key in Round 2, allowing the prover to proceed with the next measurement in the clear.

Our second contribution streamlines the ideas introduced to prove Theorem 1 in order to obtain a 2-round proof of quantumness relying on the classical hardness of the Learning-with-Errors (LWE) problem [Reg09]. Our construction has the main advantage of being simpler than existing ones in the literature, in the sense explained below. Our proof of quantumness makes use of Noisy Trapdoor Claw-Free functions (NTCFs), introduced in [BCM$^+$21] (but it does not require the NTCFs to have an "adaptive hardcore bit" property). It relies on the particular structure of the "encrypted CNOT operation" introduced in Mahadev's QFHE scheme [Mah20]. We informally state our result, but we defer the construction to the technical overview.

Theorem 2 (Informal)**.** *Assuming the classical hardness of LWE, there exists a 2-round (i.e. 4-message) proof of quantumness with the following properties:*

- *It requires only one coherent evaluation of an NTCF, and one layer of single-qubit Hadamard gates.*
- *The quantum device only needs to maintain a single qubit coherent in-between the two rounds.*

Our proof of quantumness can be seen as combining ideas from [KLVY23] and [KMCVY22, AMMW22]. It is simpler than existing proofs of quantumness in the following ways:

- *Single encrypted CNOT operation.* The 2-round proof of quantumness of KLVY is based on a QFHE scheme. Concretely, an implementation of their proof of quantumness based on Mahadev's QFHE scheme requires performing a homomorphic controlled-Hadamard gate, which requires three sequential applications of the "encrypted CNOT operation". Crucially, these three operations require computing the NTCF three times in superposition while maintaining coherence all along. Our 2-round proof of quantumness only requires a single application of the "encrypted CNOT" operation. Moreover, in KLVY, the encrypted subsystem, on which Alice's operations are applied homomorphically needs to remain entangled with a second subsystem, which is used to perform Bob's operations in the clear. In contrast, in our proof of quantumness, the encrypted CNOT operation and the subsequent operations in the clear happen on a single system (of the same size as Alice's in KLVY).
- *Single qubit coherent across rounds and 2 rounds of interaction.* Compared to the 3-round proof of quantumness of [KMCVY22], ours requires one less round of interaction. However, more importantly than the number of rounds, the protocol from [KMCVY22] requires the quantum device to keep a superposition over preimages

of the NTCF coherent in-between rounds, while waiting for the next message. In contrast, both our proof of quantumness and that of KLVY only require the quantum device to keep a *single* qubit coherent in-between rounds.

- *Simple quantum operations.* The 2-round proof of quantumness of [AMMW22] matches ours in that it requires a single coherent application of an NTCF based on LWE, and the quantum device only needs to keep a single qubit coherent in between rounds. However, their protocol requires the prover to perform single-qubit measurements in "rotated" bases coming from a set of a size that scales linearly with the LWE modulus (for which typical parameters are $\approx 10^2$ to 10^3). Their completeness-soundness gap also suffers a loss compared to ours that comes from the use of the "rotated" basis measurements.

To the best of our knowledge, the only aspect in which our proof of quantumness compares unfavourably with existing ones, e.g. [KMCVY22], is that, based on current knowledge of constructions of NTCFs, our proof of quantumness requires a construction from LWE (in order to implement the "encrypted CNOT" procedure), whereas [KMCVY22] has the flexibility that it can be instantiated using any TCF, e.g. based on Diffie-Hellman or Rabin's function. While the latter are more efficient to implement, they also generally require a larger security parameter (inverting Rabin's function is as hard as factoring, whereas breaking LWE is as hard as worst-case lattice problems). Hence, it is currently still unclear which route is closer to a realisation at scale. In particular, we note that instantiating the construction of [KMCVY22] with the $x^2 \bmod N$ function by relying on a novel multiplication algorithm leads to a very efficient quantum circuit for the prover as shown in [KMY24]. It is plausible that future improvements to our proof of quantumness might yield constructions from a broader class of hardness assumptions than LWE (e.g. Ring-LWE), which would likely yield a further improvement in concrete efficiency[9].

Some existing proofs of quantumness are non-interactive (i.e. 1-round), with security in the random oracle model [BKVV20, ACGH20, YZ22]. Likewise, our proof of quantumness can also be made non-interactive by using the Fiat-Shamir transformation [FS87] (where the computation of the hash function is classical, and does not increase the complexity of the actual quantum computation). The proof of quantumness in [YZ22] has the additional desirable property of being publicly-verifiable, although it currently seems to be more demanding than the others in terms of quantum resources.

1.2 Future Directions

- *Better contextuality compilers.* Focusing on compilers for contextuality, the following important aspect remains to be strengthened. The current compiler does not in general achieve *quantum soundness* (in the sense that there are contextuality games in which a QPT prover can do much better than the "compiled" quantum value from Theorem 1). Interestingly, recent works show that KLVY does satisfy quantum soundness for certain families of non-local games [NZ23, BGKM+23, CMM+24].

- *Oblivious Pauli pad.* We think that the new functionality of the oblivious Pauli pad (or some variant of it) has the potential to be useful elsewhere, and we leave this exploration to future work. A related question is to realise the oblivious Pauli pad in the plain model. We note that the *oblivious pad* can be constructed from the hardness of factoring if one does not require the "classical range sampling" property (see Definition 19), but we do not know whether one can have a plain model construction that satisfies Definition 19 in full (we discuss this in more detail in Section 8).

- *More efficient "encrypted CNOT".* Turning to proofs of quantumness, the broad goal is of course to simplify these even further towards experimental implementations. One concrete direction in which our proof of quantumness could be simplified is the following. Currently, we use an NTCF that supports the "encrypted CNOT" operation from [Mah20]. However, our only requirement is that the NTCF satisfies the potentially weaker property that it hides a bit in the xor of the first bit of a pair of preimages. This is because we only need to perform the "encrypted CNOT" operation once, and not repeatedly as part of a full-fledged homomorphic computation. It would be interesting to see if the weaker requirement could be achieved by simpler claw-free functions (from LWE or other assumptions, like DDH or factoring).

- *Testing other sources of quantumness.* Many other sources of quantumness have been identified in the literature, such as generalised contextuality (which allows arbitrary experimental procedures, not just projective

[9]An upcoming work by one of the authors in fact improves and instantiates our template using Ring-LWE.

measurements, and a broad class of ontological models, not just deterministic ones) [Spe05] and the Leggett-Garg experiment (which is the time analogue of Bell's experiment) [LG85, BMKG13]. These all suffer from the same limitation as contextuality, and our results raise the following question: can one construct analogous operational single-device tests for these sources of quantumness as well?

- *Testing indefinite causal order.* More ambitiously, one could try to separate quantum mechanics from more general theories such as those with indefinite causal order [CDPV13, OCB12] (i.e. theories that obey causality only locally and that may not admit a definite causal order globally). For instance, a recent result gives evidence (in the black-box model) that indefinite causal order does not yield any relevant advantage over quantum mechanics [AMP23]. Perhaps one can obtain a clear separation under cryptographic assumptions?

Acknowledgements

We are thankful to an anonymous QCrypt reviewer for their observations about the relation between the oblivious pad and non-interactive proofs of quantumness, and for pointing out the limitations of not formalising what it means for a test to be an operational test of contextuality.

We thank Ulysse Chabaud, Mauro Morales and Thomas Vidick for their feedback on early drafts of this work. We thank Yusuf Alnawakhtha, Alexandru Gheorghiu, Manasi Mangesh Shingane, and Urmila Mahadev for various helpful discussions. ASA acknowledges support from the U.S. Department of Defense through a QuICS Hartree Fellowship, IQIM, an NSF Physics Frontier Center (GBMF-1250002) and MURI grant FA9550-18-1-0161. Part of the work was carried out while ASA was visiting the Simons Institute for the Theory of Computing. AC acknowledges support from the National Science Foundation grant CCF-1813814, from the AFOSR under Award Number FA9550-20-1-0108 and from the Engineering and Physical Sciences Research Council through the Hub in Quantum Computing and Simulation grant (EP/T001062/1) and the Quantum Advantage Pathfinder (QAP) research programme (EP/X026167/1). KB is supported by A*STAR C230917003.

2 Technical Overview

In this overview, we start by briefly recalling non-local games, and introducing the notion of contextuality with some examples. We then build up towards our compiler for contextuality games, by first introducing the ideas behind the KLVY compiler, and then describing why new ideas are needed to achieve a compiler in the contextuality setting. We then describe our main novel technical tool, the oblivious pad, and how we use it to realise a compiler for contextuality games. We also discuss the main ideas in the proof. Finally, we describe how some of the new ideas can be streamlined to obtain a potentially simpler proof of quantumness. The latter can be understood without any reference to contextuality, and the interested reader may wish to skip directly to it (Section 2.6).

2.1 Non-local Games

Let A, B, X, Y be finite sets. A 2-player non-local game is specified by a predicate $\mathsf{pred} : A \times B \times X \times Y \to \{0, 1\}$ which indicates whether the players win or not, and a probability distribution $\mathcal{D}_{\text{questions}}$ over the questions, which specifies $\Pr(x, y)$ for $(x, y) \in X \times Y$. The game consists of a referee and two players, Alice and Bob, who can agree on a strategy before the game starts, but cannote communicate once the game starts. The game proceeds as follows: the referee samples questions $(x, y) \leftarrow \mathcal{D}_{\text{questions}}$, sends x to Alice and y to Bob, and receives their answers $a \in A$ and $b \in B$ respectively. Their success probability can be written as

$$\Pr(\text{win}) = \sum_{x,y,a,b} \mathsf{pred}(a, b, x, y) \Pr(a, b|x, y) \Pr(x, y) , \tag{1}$$

where the strategy used by Alice and Bob specifies $\Pr(a, b|x, y)$.

Local Hidden Variable Strategy The "local hidden variable" model allows Alice and Bob to share a classical (random) variable r, and then have their answers be arbitrary, but fixed, functions of their respective questions, i.e.

$$\Pr(a, b|x, y) = \sum_r P_A(a|x, r) P_B(b|y, r) P_R(r)$$

where P_A, P_B and P_R are probability distributions that specify Alice and Bob's strategy. We refer to the optimal winning probability achievable by local hidden variable strategies as the *classical value* of the game.

Quantum Strategy A quantum strategy allows Alice and Bob to share a state $|\psi\rangle_{AB}$, and use local measurements $M_x^A = \{M_{a|x}^A\}_a$ and $M_y^B = \{M_{b|y}^B\}_b$ to produce their answers (where the measurements can be taken to be projective without loss of generality), so that

$$\Pr(a, b|x, y) = \langle\psi| M_{a|x}^A \otimes M_{b|y}^B |\psi\rangle .$$

We refer to the optimal winning probability achievable by quantum strategies as the *quantum value* of the game.

It is well-known that there exist non-local games (like the CHSH game) where the quantum value exceeds the classical value. However, is it necessary to consider spatial separation (i.e. a tensor product structure) to observe such a "quantum advantage"? A partial answer is no: one can consider contextuality.

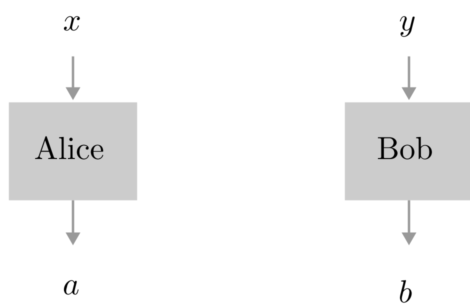


Figure 3: A two-player non-local game. Alice and Bob get (x, y) from the referee and respond with (a, b). They cannot communicate once the game starts.

2.2 Contextuality

We start with a slightly informal definition of a contextuality game (see Section 6 for a formal treatment). Let Q and A be finite sets. Let C^{all} be a set of subsets of Q. We refer to the elements of C^{all} as *contexts*. Suppose for simplicity that all subsets $C \in C^{\mathrm{all}}$ have the same size k. A *contextuality game* is specified by a predicate $\mathsf{pred} : A^k \times C^{\mathrm{all}} \to \{0, 1\}$, and a probability distribution $\mathcal{D}_{\mathrm{contexts}}$ over contexts, which specifies $\Pr(C)$ for $C \in C^{\mathrm{all}}$. The game involves a referee and a *single* player, and proceeds as follows:

- The referee samples $C = \{q_1, \ldots, q_k\} \in C^{\mathrm{all}}$ according to $\mathcal{D}_{\mathrm{contexts}}$, and sends C to the player.
- The player responds with answers $\{a_1, \ldots, a_k\} \in A^k$.

The success probability is

$$\Pr(\mathrm{win}) = \sum_{a_1, \ldots, a_k, C} \mathsf{pred}(a_1, \ldots, a_k, C) \Pr(a_1, \ldots, a_k|C) \Pr(C) , \tag{2}$$

where the strategy used by the player specifies $\Pr(a_1, \ldots, a_k|C)$.

Non-contextual strategy This is the analogue of a "local hidden variable" strategy. A *deterministic assignment* maps each question $q \in Q$ to a *fixed* answer a_q. This represents the following strategy: upon receiving the context $C = \{q_1, \ldots, q_k\} \in C^{\mathrm{all}}$, return $(a_{q_1}, \ldots, a_{q_k})$ as the answer. A strategy that can be expressed as a convex combination of deterministic assignments is referred to as *non-contextual*, i.e. the answer to a question q is independent of the context in which q is being asked.

Quantum strategy A quantum strategy is specified by a quantum state $|\psi\rangle$, and a collection of observables $\mathbf{O} = \{O_q\}_{q\in Q}$, such that, for any context $C = \{q_1, \ldots, q_k\} \in C^{\text{all}}$, the observables $O_{q_1}, \ldots, O_{q_k}$ are compatible (i.e commuting). The strategy is the following: upon receiving context $C = \{q_1, \ldots, q_k\} \in C^{\text{all}}$, measure observables $O_{q_1}, \ldots, O_{q_k}$ on state $|\psi\rangle$, and return the respective outcomes $a_1, \ldots, a_k$.

Quantum mechanics is *contextual* in the sense there are examples of games for which a quantum strategy can achieve a higher winning probability than the best non-contextual strategy. However, crucially, unlike a non-local game, a contextuality game does not directly yield an "operational test" of contextuality. The issue is that there is no clear way for the referee to enforce that the player's answer to question q is consistent across the different contexts in which q appears!

We informally describe three simple examples: the magic square game (Peres-Mermin) [Per90, Mer90, Cab01, Ara04], non-local games, and the KCBS experiment (the contextuality analogue of the Bell/CHSH experiment) [KCBS08]. We do this by directly specifying a quantum strategy first and then "deriving" the corresponding game (where the observables are just labels for the questions). In doing so, we abuse the notation slightly and identify questions with observables.

Example 1 (Peres-Mermin (Magic Square) [Per90, Mer90])**.** Consider the following set of observables

$$\mathbf{O} := \begin{array}{cccc} \{X\otimes\mathbb{I}, & \mathbb{I}\otimes Z, & X\otimes Z, & \mathbb{I} \\ \mathbb{I}\otimes X, & Z\otimes\mathbb{I}, & Z\otimes X, & \mathbb{I} \\ X\otimes X, & Z\otimes Z, & Y\otimes Y\} & \mathbb{I} \\ & & & \\ \mathbb{I} & \mathbb{I} & -\mathbb{I} & \end{array}$$

(where X, Y, Z are Pauli matrices). They satisfy the following properties: (a) they take ±1 values (i.e. they have ±1 eigenvalues), (b) operators along any row or column commute, and (c) the product of observables along any row or column equals $\mathbb{I}$, except along col_3, where it equals $-\mathbb{I}$. If we define the set of contexts by $C^{\text{all}} := \{\text{row}_1, \text{row}_2, \text{row}_3, \text{col}_1, \text{col}_2, \text{col}_3\}$, it is not difficult to see that no deterministic assignment can be such that the condition on the products is satisfied along each row and column. For instance, the assignment

$$\begin{array}{ccc} 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & 1 & ? \end{array}$$

satisfies all constraints except one: if the question mark is 1, then the condition along col_3 fails, and if it is -1 then the condition along row_3 fails. To satisfy both, somehow the value assigned to the last "observable" has to depend on the context, row_3 or col_3, in which it appears—the assignment has to be "contextual". In quantum mechanics, all of the constraints can be satisfied using the observables described. One can in fact measure these observables on an arbitrary state $|\psi\rangle$ to win with probability 1. One thus concludes that quantum mechanics is "contextual" in this sense.

2-player non-local games can be viewed as a special case of contextuality games as follows (and similarly for non-local games with more players).

Example 2 (2-player non-local games)**.** Given a quantum strategy for a non-local game (as in Section 2.1), we identify the measurements M_x^A and M_y^B with corresponding observables. Then, define $\mathbf{O} := \{M_x^A \otimes I\}_x \cup \{I \otimes M_y^B\}_y$ and $C^{\text{all}} := \{\{M_x^A \otimes I, I \otimes M_y^B\}\}_{x,y}$. It is not hard to see that the set of non-contextual strategies is the same as the set of local hidden variable strategies.

Some contextuality games can yield a separation between non-contextual and quantum strategies with even smaller quantum systems than what is possible for non-local games. The following example yields a separation using just a qutrit—a single 3-dimensional system (in contrast, non-local games require at least two qubits, i.e. dimension 4, as in the CHSH game). For contextuality 3 dimensions are necessary and sufficient [KS67].[10]

Example 3 (KCBS [KCBS08])**.** Consider a 3-dimensional vector space spanned by $\{|0\rangle, |1\rangle, |2\rangle\}$. Let $|\psi\rangle = |0\rangle$ and define five vectors

$$|v_q\rangle := \cos\theta\,|0\rangle + \sin\theta\sin\phi_q\,|1\rangle + \sin\theta\cos\phi_q\,|2\rangle\,,$$

[10] There are generalisations of contextuality [Spe05] that can give a separation with dimension 2, but we do not consider these here.

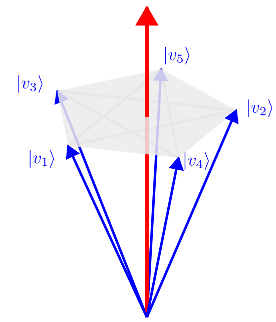


Figure 4: An illustration of the optimal quantum strategy corresponding to the KCBS Game defined in Example 3. Here the red vector denotes the quantum state $|\psi\rangle$ and the blue ones $|v_q\rangle$ correspond to projective measurements, $\Pi_q = |v_q\rangle\langle v_q|$. Consecutively indexed blue vectors, i.e. $|v_q\rangle, |v_{q+1}\rangle$, are orthogonal (indexing is periodic).

indexed by $q \in \{1, \dots, 5\}$ where $\phi_q = 4\pi q/5$ and $\cos^2\theta = \cos(\pi/5)/(1+\cos(\pi/5))$. The heads of these vectors form a pentagon with $|\psi\rangle$ at the centre, and vectors indexed consecutively are orthogonal, i.e. $\langle v_q | v_{q+1}\rangle = 0$ (where we take the indices to be periodic) as illustrated in Figure 4. Define $\mathbf{O} := \{\Pi_q\}_{q\in 1\dots 5}$ where $\Pi_q := |v_q\rangle\langle v_q|$ and $C^{\text{all}} := \{\{\Pi_1, \Pi_2\}, \{\Pi_2, \Pi_3\} \dots \{\Pi_5, \Pi_1\}\}$. The referee asks a context $C \leftarrow C^{\text{all}}$ uniformly at random from the set of all contexts and the player wins if the answer is either $(0,1)$ or $(1,0)$ (i.e. neighbouring assignments should be distinct). It is not hard to check that a non-contextual strategy wins at most with probability $4/5 = 0.8$, while the quantum strategy described above wins with probability $\frac{2}{\sqrt{5}} \approx 0.8944$.

One route towards obtaining an operational test of contextuality is to find a way to enforce that measurements on a single system happen "sequentially", i.e. they are separated in "time" (as opposed to being separated in "space", which is the case for non-local games). We describe one folklore attempt at constructing an operational test, which assumes that the device is "memoryless". This example is not essential to understanding our results, and may be skipped at first read.

The "memoryless" attempt at an operational test As denoted in Figure 5a, consider a device that takes as input a question q, produces an answer a, and then forgets the question. The referee in this case proceeds as follows:

1. Samples a context C from C^{all} with probability $\Pr(C)$.
2. Sequentially asks all the questions $q_1 \dots q_k$ in the context C.
3. Accepts the answers $a_1 \dots a_k$ if the constraint corresponding to C holds, i.e. if $\mathsf{pred}(a_1 \dots a_k, C)$ is true.

Note that the most general deterministic model for the device is one that encodes a "truth table" $\tau : Q \to A$ that maps questions to answers. Since there is no memory, no previous question can affect the way the device answers the current question. The most general quantum device, on the other hand, starts with an initial $|\psi\rangle$ and to each question, assigns an observable O_q, which is measured to obtain an answer a to the question q.

Let us work out an example to illustrate the key point. Consider again the Magic Square game from Example 1. Suppose the first question the referee asks is the value of the bottom right box of the magic square (see Figure 5b). It can then either ask questions completing the corresponding row or column. For a deterministic *memoryless* device, since a single truth table τ is being used, it is impossible to satisfy all the constraints of the magic square. However, a deterministic device *with memory* can satisfy all the constraints. This is because before answering the last question, it can learn whether the third column is being asked or the third row and thus, it can answer the last question to satisfy the constraint. Thus "classical memory", allows for classical simulation of contextuality in this test, without using any quantum effects. In fact, [KGP+11, CGGX18] even quantifies the amount of classical memory needed to simulate contextuality.

Evidently, the glaring limitation of this attempt is that there is no operational way of ensuring that a device is "memoryless". Thus, this has remained a barrier despite the numerous attempts [LLS+11, UZZ+13, JRO+16, ZKK+17, MZL+18, LMZ+18, ZXX+19, UZZ+20, WZL+22, HXA+23, LMX+23, BRV+19b, BRV+19a, XSBC24, SSA20, BCG+22]

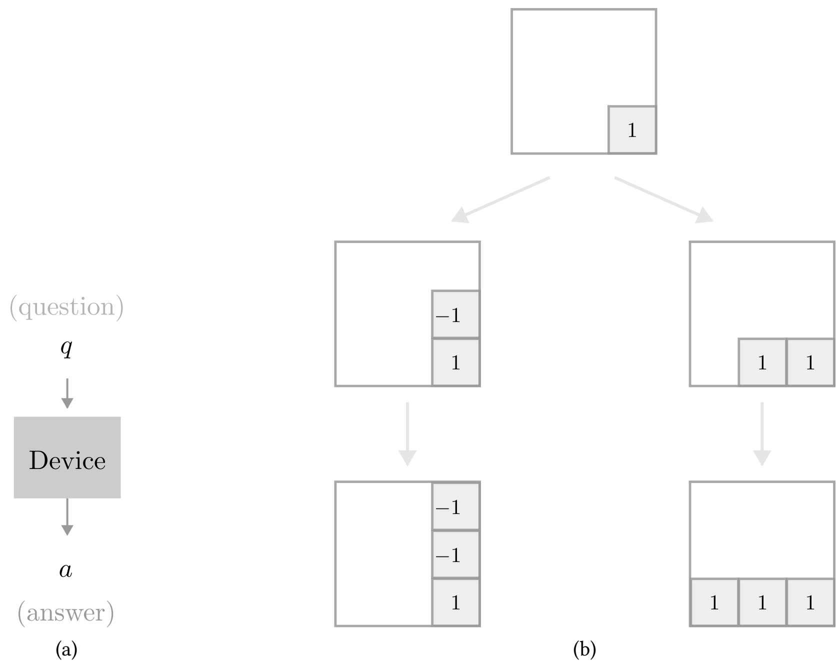


Figure 5: The folklore memoryless interpretation of contextuality: a memoryless device (left) and a memoryless operational test, illustrated using the Peres-Mermin magic square (right).

and the fact that contextuality has, in general, been a bustling area of investigation [BCG+22, WZL+22, HXA+23, LMX+23, Liu23, XSBC24].

Our construction follows the same general approach of enforcing separation in "time" instead of in "space", but is a radical departure from the attempt described above. In the subsequent sections, we only assume that the device is *computationally bounded*, and use cryptographic techniques to construct an "operational test of contextuality".

Remark 3 (Criteria for being an Operational Test of Contextuality). We have not yet formally defined what we mean by an "operational test of contextuality". Consider any test that serves as a proof of quantumness. Intuitively, we require that this test, additionally, is *faithful to some contextuality game* G, i.e. it satisfies the following two properties:

- The test involves asking the prover questions corresponding to G, possibly under some "encoding". The test is, however, allowed to involve other messages unrelated to G.
- Whether the test passes or fails is determined solely by evaluating the predicate corresponding to G, using the "decoded" questions and answers.

Formalising these requirements is a bit more involved and is deferred to Section 7. However, this intuitive notion suffices for now, as will be clear from our construction in the subsequent sections.

2.3 Quantum Fully Homomorphic Encryption (QFHE)

We informally introduce fully homomorphic encryption. We start with the classical notion.

Fully Homomorphic Encryption (FHE) A homomorphic encryption scheme is specified by four algorithms, $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$, as follows:

- Gen takes as input a security parameter 1^λ, and outputs a secret key sk.
- Enc takes as input a secret key sk and a message s, and outputs a ciphertext c. We use the notation $\mathsf{Enc}_{\mathsf{sk}} = \mathsf{Enc}(\mathsf{sk}, \cdot)$.

- Dec takes as input a secret key sk, a ciphertext c and outputs the corresponding plaintext s. We use the notation $\mathsf{Dec}_{\mathsf{sk}} = \mathsf{Dec}(\mathsf{sk}, \cdot)$.

The "homomorphic" property says that a circuit Circuit can be applied on an encrypted input to obtain an encrypted output, i.e.

- Eval takes as input a circuit Circuit, a ciphertext c, and an auxiliary input aux, and outputs a ciphertext c'. The following is satisfied. Let $c \leftarrow \mathsf{Enc}_{\mathsf{sk}}(s)$ and $c' \leftarrow \mathsf{Eval}(\mathsf{Circuit}, c, \mathsf{aux})$, then $\mathsf{Dec}_{\mathsf{sk}}(c') = \mathsf{Circuit}(s, \mathsf{aux})$.

Crucially, note that the secret key sk is not needed to apply the Eval algorithm. The security condition is the usual one: that an encryption of s should be indistinguishable from an encryption of $s' \neq s$. If Eval supports evaluation for the class of all polynomial-size circuits (in the security parameter), then the scheme is said to be *fully* homomorphic.[11]

Quantum Fully Homomorphic Encryption (QFHE) For this overview, it suffices to take a QFHE scheme to be the same as an FHE scheme, except that it allows messages and auxiliary inputs to be quantum states, and circuits to be quantum circuits. The QFHE schemes that are relevant to our work [Mah20, Bra18] satisfy the following additional properties.

1. *Classical encryption/decryption.*
   Gen is a classical algorithm, while Enc, Dec become classical algorithms when their inputs are classical. In particular, this means that classical inputs are encrypted into classical ciphertexts. This property is essential when the scheme is deployed in protocols involving classical parties, as will be the case here.[12]

2. *Locality is preserved.*
   Consider an arbitrary bipartite state $|\psi\rangle_{AB}$ and let M^A and M^B be circuits acting on registers A and B respectively with a measurement at the end. The property requires that correlations between the measurement outcomes from M^A and M^B should be the same in the following two cases:
   (i) Register A is encrypted, M_A is applied using Eval, and the result is decrypted. Let a be the decrypted outcome. M^B is applied to register B. Let b be the outcome.
   (ii) Apply M^A on register A and M^B on register B. Let a and b respectively be the outcomes.

3. *Form of Encryption.*
   Encryption of a state $|\psi\rangle$ takes the form $(X^x Z^z |\psi\rangle, \widehat{xz}) \leftarrow \mathsf{Enc}_{\mathsf{sk}}(|\psi\rangle)$, where X^x applies a Pauli X to the i-th qubit based on the value of x_i, and Z^z is defined similarly. $\widehat{xz}$ is a classical FHE encryption of the pad xz.

All known constructions of QFHE schemes satisfying property 1 also satisfy properties 2 and 3. The KLVY compiler, as one might guess, relies on property 2. Our compiler relies on property 3.[13]

2.4 The KLVY Compiler

Consider a two-player non-local game specified by question and answer sets X, Y, A, B, a predicate $\mathsf{pred} : A \times B \times X \times Y \to \{0, 1\}$ and a distribution over the questions $\mathcal{D}_{\text{questions}}$ that specifies $\Pr(x, y)$. The KLVY compiler takes this non-local game as input and produces the following single-prover game, where the verifier proceeds as follows:

- **Round 1.** Sample $(x, y) \leftarrow \mathcal{D}_{\text{questions}}$, and a secret key sk for a QFHE scheme. Send an encryption c_x of "Alice's question", and get an encrypted answer c_a.

- **Round 2.** Send "Bob's question" y, and get his answer b in the clear. Decrypt Alice's response using sk, and accept if $\mathsf{pred}(a, b, x, y) = 1$.

The honest prover prepares the entangled state $|\psi\rangle_{AB}$ corresponding to the optimal quantum strategy for the non-local game. It uses QFHE's Eval algorithm on subsystem A to answer question x according to Alice's optimal strategy, and answers question y in the clear using Bob's optimal strategy on subsystem B. More formally, they proceed as in Figure 6.

[11] While homomorphic schemes for restricted families of circuits have been known for some time, a "fully homomorphic scheme" was only discovered somewhat recently in [Gen09].

[12] The first schemes to satisfy this property appeared in the breakthrough works [Mah20, Bra18].

[13] Strictly speaking, we require the FHE scheme in property 3 to be public-key. This is used to establish the faithfulness condition (detailed in Definition 18 and Section 3.2). We neglect this here for readability.

(Honest Quantum) Prover		Verifier
Prepare $|\psi\rangle_{AB}$ corresponding to the non-local game's optimal quantum strategy.		$\mathsf{sk} \leftarrow \mathsf{Gen}(1^\lambda)$ $(x, y) \leftarrow \mathcal{D}_{\text{questions}}$ $c_x \leftarrow \mathsf{Enc}_{\mathsf{sk}}(x)$
	$\xleftarrow{c_x}$	
Homomorphically compute the answer to x using "Alice's strategy" on system A. Let c_a be the encrypted answer.		
	$\xrightarrow{c_a}$	
	$\xleftarrow{y}$	
Compute the answer to y using "Bob's strategy" (in the clear) on system B. Let the answer be b.		
	$\xrightarrow{b}$	
		$a = \mathsf{Dec}_{\mathsf{sk}}(c_a)$ Accept if $\mathsf{pred}(a, b, x, y) = 1$.

Figure 6: The KLVY compiler [KLVY23] is illustrated above: it takes any two-player non-local game and converts it into a proof of quantumness.

Theorem ([KLVY23], informal)**.** *Consider a two-player non-local game with classical and quantum values ω_c and ω_q respectively. The corresponding KLVY-compiled single-prover game satisfies the following:*

- *(Completeness) The QPT prover described above makes the verifier accept with probability $\omega_q - \mathsf{negl}(\lambda)$.*
- *(Soundness) Every PPT prover makes the verifier above accept with probability at most $\omega_c + \mathsf{negl}(\lambda)$.*

Here negl *are (possibly different) negligible functions.*

For completeness, we briefly outline the general compiler for non-local games with $k > 2$ players. To compile games with k players, KLVY generalises the procedure above as follows: the compiled game consists of k rounds–each round consisting of a question and an answer (so $2k$ messages in total). The first $k-1$ questions are encrypted (using $k-1$ independent random QFHE keys) and the last question is asked in the clear.

Let us take a moment and build some intuition about the KLVY compiler. At first, one might think that it would be even more secure to also encrypt y. However, as we remarked in the introduction, this is not the case: since the prover can compute any desired answer b using x, a and y under the hood of the QFHE, it can ensure that b is such that $\mathsf{pred}(a, b, x, y) = 1$ (for any non-trivial choice of pred). Instead, the key observation of KLVY is that the verifier is testing the correlation between encrypted answers and answers in the clear, and it turns out that a quantum prover can produce stronger correlations than any PPT prover can. How is this intuition formalised? The key idea is that, since a PPT prover is classical, one can rewind the PPT prover to obtain answers to *all* possible second round questions. This is equivalent to obtaining Bob's entire assignment of answers to questions (corresponding to a fixed encrypted question and answer for Alice from the first round). Suppose for a contradiction that such a PPT prover wins with probability non-negligibly greater than the classical value. Then, it must be that Bob's entire assignment is non-trivially correlated to "Alice's encrypted question". This information can thus be used to obtain a non-negligible advantage in guessing the encrypted question, breaking security of the QFHE scheme.

Now that we understand the KLVY construction and the key insight behind their proof, we will discuss barriers to extending these ideas to contextuality, and our approach to circumvent these.

2.5 Contribution 1 | A Computational Test of Contextuality

For simplicity, let us first restrict to contextuality games with contexts of size 2.

Attempts at extending KLVY to contextuality. Let us consider compilers for contextuality games where the verifier proceeds in an analogous way as the verifier in the KLVY compiler:

- **Round 1.** Sample a context $C = (q_1, q_2) \leftarrow \mathcal{D}_{\text{contexts}}$. Send a QFHE encryption c_{q_1} of q_1 and receive as a response an encryption c_{a_1} of a_1.
- **Round 2.** *There is no clear way to proceed. Three natural approaches are listed below.*

The honest prover's state after Round 1 is encrypted and has the form $(X^x Z^z \left|\psi_{(q_1,a_1)}\right\rangle, \widehat{xz})$ where $\widehat{xz}$ denotes a classical encryption of the strings xz (using Property 3 of the QFHE scheme), as detailed below in Figure 7.

Here are three natural approaches for how to proceed, and how they fail.

1. *Proceed just as KLVY: Ask question q_2 in the clear.* This does not work because, unlike in the original KLVY setup, there is no analogue of system B which is left in the clear. Here the prover holds an encrypted state so it is unclear how q_2 can be answered with any non-trivial dependence on the state under the encryption.
2. *Ask the second question also under encryption.* If the same key is used for encryption, then, as we argued for KLVY, the predicate of the game can be satisfied by computing everything homomorphically. If the keys are independent, then we essentially return to the problem in item 1.
3. *Reveal the value of the (classically) encrypted pad $\widehat{xz}$ and ask question q_2 in the clear.* This has a serious issue: the prover can simply ask for the encrypted pad corresponding to c_{q_1} and thereby learn q_1 (or at least some bits of q_1). Once q_1 is learned, again, the predicate of the game can be trivially satisfied.

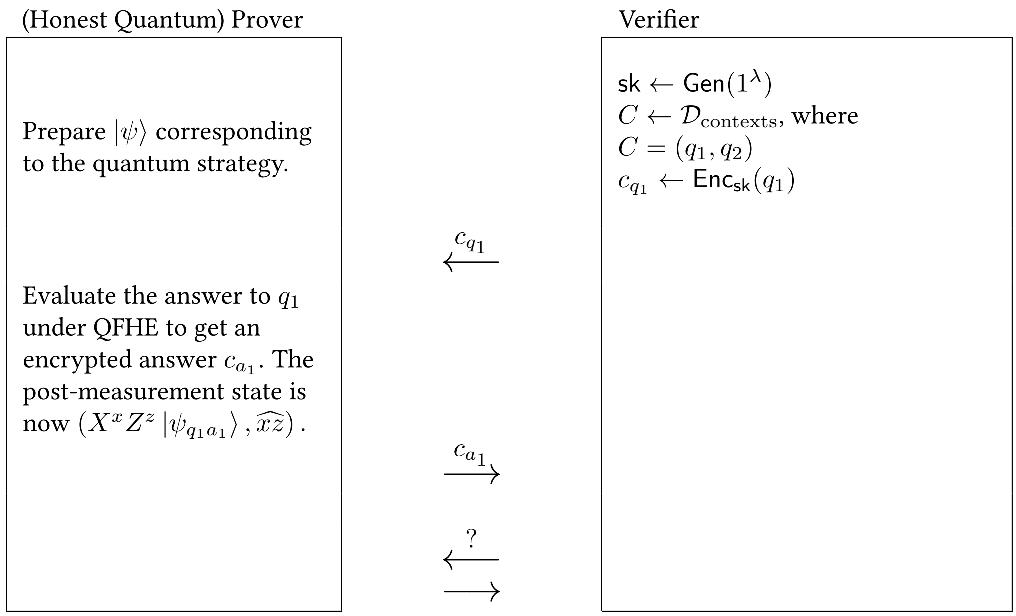


Figure 7: The figure illustrates that, a priori, it is unclear how to generalise KLVY to compile contextuality games.

The Oblivious Pauli Pad As mentioned in Section 1, our compiler relies on a new cryptographic primitive, that we introduce to circumvent the barriers described above. Here, it helps to be a bit more formal. Let $\mathbf{U} := \{U_k\}_{k \in K}$ be a group of unitaries acting on the Hilbert space $\mathcal{H}$. We take this group to be the set of Paulis $\{X^x Z^z\}_{xz}$, as this makes the primitive compatible with the the form of the QFHE scheme that we will employ later in our compiler. Nonetheless, we use the general notation $\{U_k\}_{k \in K}$, as it simplifies the presentation. We define the *oblivious* $\mathbf{U}$ *pad* as follows.

The *oblivious* $\mathbf{U}$ *pad* is a tuple of algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ where Gen and Dec are PPT.[14] Let $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ be the public and the secret keys generated by Gen. Encryption takes the form $(U_k |\psi\rangle, s) \leftarrow \mathsf{Enc}_{\mathsf{pk}}(|\psi\rangle)$, where $k = \mathsf{Dec}_{\mathsf{sk}}(s)$. Notice the similarity with the post-measurement state in the discussion above (we will return to this in a moment). The security requirement is that no PPT algorithm can win the following security game with probability non-negligibly greater than $1/2$. We depict the oblivious pad primitive in Figure 8.

The security game formalises the intuition that no PPT prover can distinguish between the correct "key" k_1 and a uniformly random "key" k_0. We emphasize, in words, the two distinctive features of this primitive:

- By running Enc, a QPT prover can obtain, given a state $|\psi\rangle$, an encryption of the form $(U_k |\psi\rangle, s)$, where $k = \mathsf{Dec}_{\mathsf{sk}}(s)$.
- There is no way for a PPT prover, given pk, to produce an "encryption" s, for which it has non-negligible advantage at guessing $\mathsf{Dec}_{\mathsf{sk}}(s)$.

We describe informally how to instantiate the primitive in the random oracle model assuming noisy trapdoor claw-free functions (the detailed description is in Algorithm 2). The construction builds on ideas from [BKVV20].

The key idea is the following. Let f_0, f_1 be a Trapdoor Claw-Free function pair. We take $\mathsf{pk} = (f_0, f_1)$, and sk to be the corresponding trapdoor. Then, $\mathsf{Enc}_{\mathsf{pk}}$ is as follows:

(i) On input a qubit state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$, evaluate f_0 and f_1 in superposition, controlled on the first qubit, and measure the output register. This results in some outcome y, and the leftover state $(\alpha |0\rangle |x_0\rangle + \beta |1\rangle |x_1\rangle)$, where $f(x_0) = f(x_1) = y$.

(ii) Compute the random oracle "in the phase", to obtain $((-1)^{H(x_0)} \alpha |0\rangle |x_0\rangle + (-1)^{H(x_1)} \beta |1\rangle |x_1\rangle)$. Measure the second register in the Hadamard basis. This results in a string d, and the leftover qubit state

$$|\psi_Z\rangle = Z^{d \cdot (x_0 \oplus x_1) + H(x_0) + H(x_1)} |\psi\rangle \, .$$

[14] The formal definition of the oblivious pad involves a fourth PPT algorithm Samp which for clarity is deferred to Section 8.

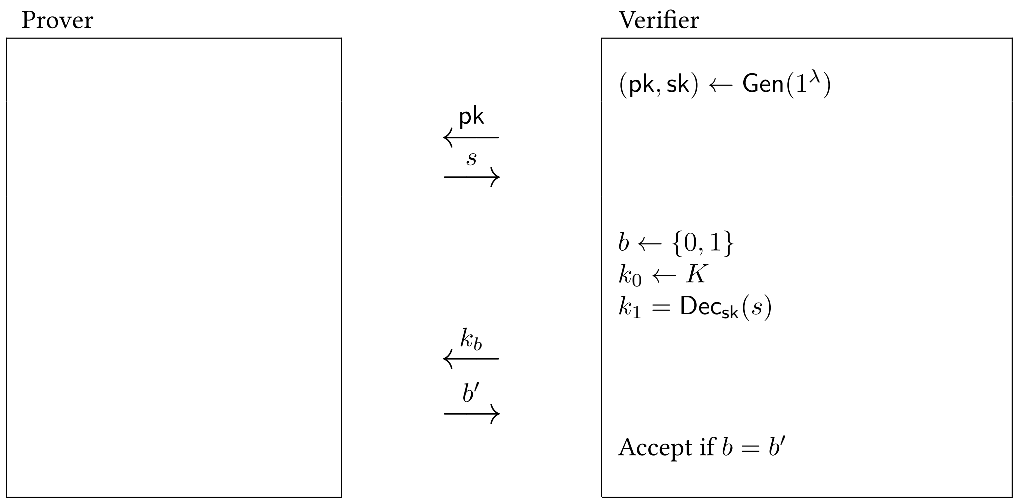


Figure 8: The security game for the oblivious pad.

(iii) Repeat steps (i) and (ii) on $|\psi_Z\rangle$, but *in the Hadamard basis*! This results in strings y' and d', as well as a leftover qubit state $|\psi_{XZ}\rangle =$

$$X^{d'\cdot(x'_0 \oplus x'_1)+H(x'_0)+H(x'_1)} Z^{d\cdot(x_0 \oplus x_1)+H(x_0)+H(x_1)} |\psi\rangle \ ,$$

where x'_0 and x'_1 are the pre-images of y'.

Notice that the leftover qubit state $|\psi_{XZ}\rangle$ is of the form $X^x Z^z |\psi\rangle$ where x, z have the following two properties: (a) a verifier in possession of the TCF trapdoor can learn z and x given respectively y, d and y', d', and (b) no PPT prover can produce strings y, d as well as predict the corresponding bit z with non-negligible advantage (and similarly for x). Intuitively, this holds because a PPT prover that can predict z with non-negligible advantage must be querying the random oracle at *both* x_0 and x_1 with non-negligible probability. By simulating the random oracle (by lazy sampling, for instance), one can thus extract a claw x_0, x_1 with non-negligible probability, breaking the claw-free property.

Our Compiler We finally describe our contextuality game compiler. As mentioned in the introduction, our strategy is still to ask the first question under QFHE encryption and the second question in the clear, with the following crucial difference: the prover is first asked to "re-encrypt" the post-measurement state using the *oblivious pad* functionality (from here one referred to as OPad), and only *after that* the verifier reveals to the prover how to "decrypt" the state, in order to proceed to round 2.

The key idea is easy to state, once the notation is clear. To this end, recall that Property 3 of a QFHE scheme ensures that the encryption of a quantum state $|\psi''\rangle$ has the form

$$(U_{k''} |\psi''\rangle \, , \hat{k}'') \tag{3}$$

where $\hat{k}''$ denotes a classical encryption of k'' (the reason why we use double primes will become clear shortly), and $U_{k''}$ is an element of the Pauli group. Note that using the secret key of the QFHE scheme, one can recover k'' from $\hat{k}''$. Further, let the optimal quantum strategy for the underlying contextuality game consist of state $|\psi\rangle$ and observables $\{O_q\}$. Finally, denote by $|\psi_{a,q}\rangle$ the post-measurement state arising from measuring $|\psi\rangle$ using O_q and obtaining outcome a.

We are now ready to describe our compiler. We first explain it in words and subsequently give a more formal description for clarity. In both cases, we highlight the conceptually new parts in blue.

- **Round 1** *Ask the encrypted question, and have the prover re-encrypt its post-measurement state using* OPad. The verifier samples keys for the QFHE scheme and for the OPad. It samples a context $C \leftarrow \mathcal{D}_{\text{contexts}}$ and then uniformly samples questions q_1, q_2 from the context C (note that $q_1 = q_2$ with probability $1/2$ since, for simplicity, we are considering contexts of size 2.)

- **Message 1** The verifier sends the QFHE encryption c_{q_1} of the first question q_1, together with the public key of the OPad.

The honest quantum prover obtains the encrypted answer c_{a_1} by measuring, under the QFHE encryption, the state $|\psi\rangle$ using the observable O_{q_1}. It now holds a state of the form in Equation (3), with $|\psi''\rangle = |\psi_{a_1,q_1}\rangle$. Using the public key of the OPad, the prover applies OPad.Enc to the encrypted post-measurement state, $U_{k''}|\psi_{a_1,q_1}\rangle$, to obtain the "re-encrypted" quantum state, $U_{k'}U_{k''}|\psi''\rangle$, where $U_{k'}$ was applied by the OPad, together with a classical string s' that encodes k'. This step is critical to the security of the protocol and is discussed in more detail shortly. Crucially, note that both $U_{k'}$ (coming from the OPad) and $U_{k''}$ (coming from the QFHE) are Paulis.

- **Message 2** The prover sends the QFHE encrypted answer c_{a_1}, together with the two strings $\hat{k}''$ and s'.

- **Round 2** *Remove the overall encryption, and proceed in the clear.* The verifier recovers k' from s' (using the secret key of the OPad) and k'' from $\hat{k}''$ (using the secret key of the QFHE scheme). It then computes k satisfying $U_k = U_{k'}U_{k''}$.

- **Message 3** The verifier sends the second question q_2 together with k as computed above.

The prover measures its quantum state $U_{k'}U_{k''}|\psi''\rangle = U_k|\psi_{a_1,q_1}\rangle$, using observable $U_k O_{q_2} U_k^\dagger$ to obtain an outcome a_2.

- **Message 4** The prover sends a_2.

The verifier decrypts c_{a_1} to recover a_1 using the secret key of the QFHE scheme. If $q_1 = q_2$, it accepts if the answers match, i.e. $a_1 = a_2$. If $q_1 \neq q_2$, it accepts if the predicate is true, i.e. $\mathsf{pred}(a_1, a_2, q_1, q_2) = 1$.

We summarise our compiler in Figure 9. Since we now have Gen, Enc, Dec algorithms for both OPad and QFHE, we use prefixes such as OPad.Enc to refer to Enc associated with OPad to avoid confusion.

Our compiler satisfies the following, assuming the underlying QFHE and oblivious pad are secure. We first state a special case of our general result (which is stated later in Theorem 4).

Theorem (restatement of Theorem 1). *Consider a contextuality game* G *with contexts of size* 2. *Let* valNC *and* valQu *be its non-contextual and quantum values respectively. The compiled game (as described above) is faithful to* G. *In particular, it satisfies the following:*

- *(Completeness) The QPT prover described above wins with probability* $\frac{1}{2}(1 + \mathrm{valQu}) - \mathsf{negl}(\lambda)$.
- *(Soundness) Any PPT prover wins with probability at most* $\frac{1}{2}(1 + \mathrm{valNC}) + \mathsf{negl}(\lambda)$.

Here negl *denote (possibly different) negligible functions.*

Proof sketch: The faithfulness condition as discussed in Remark 3 is evidently satisfied by the compiled game.[15] Suppose $\mathcal{A}$ is a PPT algorithm that wins with probability non-negligibly greater than $\frac{1}{2}(1 + \mathrm{valNC})$. Observe that one can associate a "deterministic assignment" corresponding to $\mathcal{A}$, conditioned on some fixed first round messages, as follows: simply rewind $\mathcal{A}$ to learn answers to all possible second round questions, obtaining an assignment $\tau : Q \to A$, mapping questions to answers. Let us write τ_{q_1} to make the dependence of the assignment on the first question more explicit (note that the assignment depends on the encrypted question c_{q_1} as well as the encrypted answer c_{a_1}). For the purpose of this overview, suppose also that $\mathcal{A}$ is consistent, i.e. if $q_1 = q_2$, then $a_1 = a_2$ (note that this in particular ensures that, when $q_1 = q_2$, $\mathcal{A}$ wins with probability 1. One can show that an adversary that is not consistent can be turned into an adversary that is consistent and wins with at least the same probability). Now, to win with probability more than $\frac{1}{2}(1 + \mathrm{valNC})$, it must be that the τ_{q_1}'s are different for different q_1's. Otherwise, $\mathcal{A}$'s strategy is just a convex combination of deterministic assignments and this by definition cannot do better than valNC when $q_1 \neq q_2$. But if the distribution over τ_{q_1}s and $\tau_{q_1'}$s is different for at least some $q_1 \neq q_1'$, then one is able to distinguish QFHE encryptions of q_1 from those of q_1'. Thus, as long as the QFHE scheme is secure, no PPT algorithm can win with probability non-negligibly greater than $\frac{1}{2}(1 + \mathrm{valNC})$.

[15] The formal notion, as stated later in Section 7, is also satisfied but the details are deferred.

(Honest Quantum) Prover

Verifier

(Honest Quantum) Prover		Verifier
		$\mathsf{sk} \leftarrow \mathsf{QFHE.Gen}(1^\lambda)$ $C \leftarrow \mathcal{D}_{\text{contexts}}$ $q_1 \leftarrow C$ $q_2 \leftarrow C$ $c_{q_1} \leftarrow \mathsf{QFHE.Enc}_{\mathsf{sk}}(q_1)$ $(\mathsf{OPad.sk}, \mathsf{OPad.pk}) \leftarrow \mathsf{OPad.Gen}(1^\lambda)$
Prepare $\lvert\psi\rangle$ corresponding to the quantum strategy.		
	$\xleftarrow{(c_{q_1}, \mathsf{OPad.pk})}$	
Evaluate the answer to q_1 under QFHE to get an encrypted answer c_{a_1}. The post-measurement state is now $\left(U_{k''}\lvert\psi_{q_1a_1}\rangle, \widehat{k}''\right)$ and now apply the OPad to the post-measurement state, i.e. $(U_{k'}U_{k''}\lvert\psi_{q_1a_1}\rangle, s') \leftarrow \mathsf{OPad.Enc}_{\mathsf{OPad.pk}}(U_{k''}\lvert\psi_{q_1a_1}\rangle)$.		
	$\xrightarrow{(c_{a_1}, \hat{k}'', s')}$	
		Using the secret keys, find k such that $U_k = U_{k'}U_{k''}$
	$\xleftarrow{(q_2, k)}$	
Measure $U_k O_{q_2} U_k^\dagger$		
	$\xrightarrow{a_2}$	
		Decrypt c_{a_1} to learn a_1. If $q_1 = q_2$, accept if $a_1 = a_2$. If $q_1 \neq q_2$, accept if $\mathsf{pred}(a_1, a_2, q_1, q_2) = 1$.

Figure 9: Our contextuality game compiler. It takes a contextuality game given by $(Q, A, C^{\text{all}}, \mathsf{pred}, \mathcal{D}_{\text{contexts}})$, a QFHE scheme and an OPad and returns an operational test of contextuality.

In the above sketch, we glossed over the very important subtlety that, in order to obtain the truth table τ, the reduction needs to provide as input to $\mathcal{A}$ the "correct" decryption key k (as the verifier does in the third message of our compiled game, where k is such that $U_k = U_{k'}U_{k''}$). However, the reduction only sees *encryptions* of k' and k''. So, how does it compute k without the secret keys? Crucially, this is where the OPad comes into play—it allows the reduction to instead use an independent uniformly random k (not necessarily the "correct" one) when constructing the reduction that breaks the security of the QFHE scheme. The fact that such a k is computationally indistinguishable from the correct one (from the point of view of the prover $\mathcal{A}$) follows precisely from the security of the OPad.

General Compilers The compiler we described earlier handles contextuality games with contexts of size 2. How does one generalise it to contexts of arbitrary size? Unlike for KLVY, it is not entirely clear what the "correct" way is here.

We design two compilers (which seem incomparable). The first compiler applies universally to all contextuality games. The second applies primarily to contextuality games where the quantum value is 1 (for instance, it works for the magic square but not for KCBS). Notably, both compilers are 4-message protocols.

- $(|C|-1,1)$ compiler:
  - Round 1: Ask $|C|-1$ questions under QFHE.
  - Round 2: Ask 1 uniformly random question in the clear. If the question was already asked in Round 1, check consistency. Otherwise, check the predicate.
- $(|C|,1)$ compiler
  - Round 1: Ask all $|C|$ questions under QFHE.
  - Round 2: Ask 1 uniformly random question in the clear, and check consistency with the questions asked in Round 1.

By now, one would be wary of guessing that asking more questions under QFHE is going to improve the security of the protocol. Indeed, the $(|C|-1,1)$ compiler (which reduces to the one we discussed above for $|C|=2$) is the universal one. We show the following.

Theorem 4 (informal)**.** *Consider an arbitrary contextuality game* G*, and let* valNC *and* valQu *be its non-contextual and quantum values respectively. The compiled game, obtained via the* $(|C|-1,1)$ *compiler, is faithful to* G *and satisfies the following:*

- *(Completeness) There is a QPT prover that wins with probability* $1-\frac{1}{|C|}+\frac{\mathrm{valQu}}{|C|}-\mathsf{negl}$.
- *(Soundness) PPT provers win with probability at most* $1-\frac{1}{|C|}+\frac{\mathrm{valNC}}{|C|}+\mathsf{negl}$.

The compiled game, obtained via the $(|C|,1)$ *compiler, is also faithful to* G*, and satisfies the following:*

- *(Completeness) There is a QPT prover that wins with probability* valQu.
- *(Soundness) PPT provers win with probability at most* $1-\mathsf{const}_1+\mathsf{negl}$*, where* $\mathsf{const}_1=\min_{C\in C^{\mathrm{all}}}\frac{\Pr(C)}{|C|}$ *(this is constant in the sense that it is independent of the security parameter), and* $\Pr(C)$ *denotes the probability of sampling the context* C.

Here, negl *are (possibly different) negligible functions.*

We make some brief remarks about the two compilers and defer the details to the main text.

- The $(|C|,1)$ compiler is not universal because, for instance, when applied to KCBS, there is no gap between the PPT and the QPT prover's winning probabilities. In fact, there is a PPT algorithm[16] that does better than the honest quantum strategy. Yet, the compiler does apply to the magic square game, for instance, because $\mathsf{const}_1<1$ and $\mathrm{valQu}=1$. In fact, for the magic square game, this compiler gives *a better completeness-soundness gap* than the $(|C|-1,1)$ compiler.

[16] Assuming that Eval is PPT if the circuit and input are classical.

- The $(|C| - 1, 1)$ compiler is universal in the sense that, when applied to any contextuality game with a gap between non-contextual and quantum value, the compiled game will have a constant gap between completeness and soundness. However, the resulting gap is sometimes smaller compared to the previous compiler. It is unclear if one can do better than this, with or without increasing the number of rounds, while preserving universality.

2.6 Contribution 2 | An Even Simpler Proof of Quantumness

Our proof of quantumness, like many of the existing ones in the literature, is based on the use of Trapdoor Claw-Free Functions (TCF). In our protocol, these are used to realize an "encrypted CNOT" functionality, which is the central building block of Mahadev's QFHE scheme [Mah20]. The "encrypted CNOT" functionality allows a prover to homomorphically apply the gate CNOT^a, while holding a (classical) encryption of the bit a. Formally, our protocol uses Noisy Trapdoor Claw-Free Functions (NTCF, defined formally in Definition 7), but here we describe our scheme using regular TCFs for simplicity.

The proof of quantumness Our 2-round proof of quantumness is conceptually very simple. It can be viewed as combining and distilling ideas from the proofs of quantumness in [KLVY23], [KMCVY22] and our contextuality compiler. We provide an informal description here, and we defer a formal description to Part I. At a high level, it can be understood as follows:

- **Round 1**: *Delegate the preparation of a uniformly random state in* $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$*, unknown to the prover.*

  The verifier samples a bit a uniformly at random.

  - **Message 1**: The verifier sends an appropriate encryption of a to the prover (and holds on to the corresponding secret key).

  The honest prover prepares the two-qubit state $|+\rangle|0\rangle$, along with auxiliary registers required to perform an "encrypted CNOT" operation. It then performs an "encrypted CNOT" operation (from the first qubit to the second), i.e. homomorphically applies CNOT^a, followed by a measurement of the second (logical) qubit.

  - **Message 2**: The prover sends all measurement outcomes to the verifier.

  Since a CNOT gate can be thought of as a deferred measurement in the standard basis, we can equivalently think of the prover's operations as performing an "encrypted measurement" of the first qubit, where the first qubit is being measured or not based on the value of a. Note that, after having performed these operations, the prover holds a *single* qubit. Thanks to the specific structure of the "encrypted CNOT" operation from [Mah20], the resulting "post-measurement" qubit state is encrypted with a Quantum One-Time Pad, and is either:

  - $|+\rangle$ or $|-\rangle$, if $a = 0$ (i.e. no logical CNOT was performed)
  - $|0\rangle$ or $|1\rangle$, if $a = 1$ (i.e. a logical CNOT was performed)

  All in all, at the end of round 1, the honest prover holds a uniformly random state in $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$, i.e. a BB84 state. This state is known to the verifier, who possesses a and the secret key. From here on, the protocol no longer uses any encryption, and everything happens "in the clear".

- **Round 2**: *Ask the prover to perform "Bob's CHSH measurement".*

  The astute reader may notice that the qubit held by the prover after Round 1 is distributed identically to "Bob's qubit" in a CHSH game where Alice and Bob perform the optimal CHSH strategy. More precisely, if one imagines that Alice has received her question and performed her corresponding optimal CHSH measurement (which is either in the standard or Hadamard basis), the leftover state of Bob's qubit is a uniformly random state in $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$, where the randomness comes both from the verifier's question (which in our protocol corresponds to the bit a), and Alice's measurement outcome.

  - **Message 3**: The verifier sends a uniformly random bit b to the prover (corresponding to Bob's question in a CHSH game).

  The prover performs Bob's optimal CHSH measurement corresponding to question b.

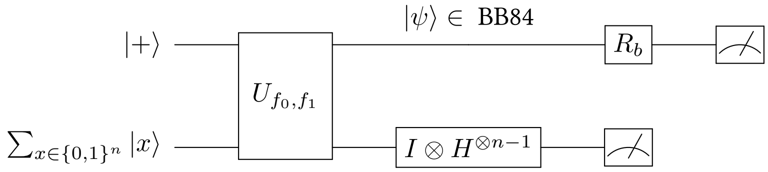


Fig. 10: The prover's circuit. Here, (f_0, f_1) is a pair of trapdoor claw-free functions (with inputs of size n). U_{f_0,f_1} denotes the $(n+1)$-qubit unitary that coherently computes f_0 in the last n qubits if the first qubit is $|0\rangle$, and computes f_1 otherwise. The circuit starts by preparing $|+\rangle$ in the first qubit, and a uniform superposition over all inputs in the next n qubits. The circuit then applies U_{f_0,f_1} (note that we are omitting auxiliary work registers that are required to compute U_{f_0,f_1}), followed by a layer of Hadamard gates on the last $n-1$ qubits. Then, the last n qubits are measured. As a result, the leftover qubit $|\psi\rangle$ in the first register is now a BB84 state (which one it is depends on f_0, f_1, and the measurement outcome). As its second message, the verifier sends a bit b, and the prover applies the rotation R_b defined as follows: $|0\rangle \overset{R_b}{\mapsto} \cos((-1)^b\pi/8)\,|0\rangle + \sin((-1)^b\pi/8)\,|1\rangle$ and $|1\rangle \overset{R_b}{\mapsto} -\sin((-1)^b\pi/8)\,|0\rangle + \cos((-1)^b\pi/8)\,|1\rangle$. Finally, the prover measures the qubit in the standard basis.[17]

- **Message 4**: The prover returns the measurement outcome to the verifier.

The verifier checks that the corresponding CHSH game is won.

In Figure 10, we show the circuit for the honest quantum prover in our proof of quantumness.

Soundness An efficient quantum prover can efficiently pass this test with probability $\cos^2(\frac{\pi}{8}) \approx 0.85$, while an efficient classical prover can pass this test with probability at most $3/4$. The proof of classical soundness is fairly straightforward. In essence, a classical prover can be rewound to obtain answers to *both* of the verifier's possible questions in Message 3. If the classical prover passes the test with probability $3/4 + \delta$ (which is on average over the two possible questions), then the answers to both questions together must reveal some information about the encrypted bit a (this is a simple consequence of how the CHSH winning conditions is defined). In particular, such a classical prover can be used to guess a with probability $\frac{1}{2} + 2\delta$. This breaks the security of the encryption, as long as δ is non-negligible. We defer the reader to Section 5.2 for more details.

Putting the ideas in perspective We have already discussed in Section 1.1 how our proof of quantumness compares to existing ones in terms of efficiency. Here, we focus on how our proof of quantumness compares conceptually to [KLVY23] and [KMCVY22]:

- In [KLVY23], the prover is asked to create an entangled EPR pair, of which the first half is encrypted, and the second half is in the clear. Then, the prover is asked to perform Alice's ideal CHSH measurement homomorphically on the first half, and Bob's CHSH measurement in the clear on the second half. Our proof of quantumness departs from this thanks to two observations:
  - By leveraging the structure of the "encrypted CNOT operation" from [Mah20], the post-measurement state from Alice's homomorphic measurement can be re-used *in the clear* (precisely because the verifier knows what the state is, but the prover does not). So the initial entanglement is not needed. This idea is also the starting point for our contextuality compiler from Section 2.5, although for the latter we take this idea much further: we find a way to give the prover the ability to decrypt the leftover state without giving up on soundness. Our proof of quantumness is a baby version of this idea: it leverages the fact that the leftover encrypted state has a special form, namely it is a BB84 state.
  - In order to setup a "CHSH-like correlation" between the verifier and the leftover qubit used by the prover in Round 2, one does not need to compile the CHSH game in its entirety. This compilation, even for the simple CHSH game requires the prover to perform an "encrypted controlled-Hadamard" operation (because Alice's ideal CHSH measurements are in the standard and Hadamard bases). The latter requires

[17] The simplified description of the proof of quantumness before this figure is slightly inaccurate: it states that the prover prepares starts by preparing the two-qubit state $|+\rangle\,|0\rangle$. Technically, the prover only needs to prepare $|+\rangle$, and the role of the second qubit is performed by the first qubit of the pre-image register (which is initialised as a uniform superposition).

three sequential “encrypted CNOT” operations. Instead, our observation is that one can setup this CHSH-like correlation more directly, as we do in Round 1 of our proof of quantumness.

- From a different point of view, our proof of quantumness can also be viewed as a simplified version of [KMCVY22]. Indeed, observation (ii) is inspired by the proof of quantumness in [KMCVY22], which introduces the idea of a “computational” CHSH test. One can interpret [KMCVY22] as setting up an “encrypted classical operation”, akin to an “encrypted CNOT”, that either entangles two registers or does not, ultimately having the effect of performing an “encrypted measurement”. This is achieved via an additional round of interaction. Our proof of quantumness can be thought of as zooming in on this interpretation, and finding a direct way to achieve this without the additional interaction.

3 Preliminaries

Section 3.1 sets up some notation, Section 3.2 formally introduces QFHE, and Section 3.3 formally introduces trapdoor-claw free functions.

3.1 Notation

- For mixed state ρ and σ, we write $\rho \approx_\epsilon \sigma$ to mean that ρ and σ are at most ϵ-far in trace distance, i.e. $\frac{1}{2}\mathrm{tr}(|\rho - \sigma|) \leq \epsilon$.
- Let X be the Pauli X matrix. For a string $x \in \{0,1\}^n$, we write X^x to mean $\otimes_{i=1}^n X^{x_i}$. We use a similar notation for Pauli Z.
- We denote the spectrum of an observable O by $\mathsf{Spectr}(O)$ and the support of a function f by $\mathsf{Supp}(f)$.
- We use the abbreviations PPT and QPT for probabilistic polynomial time and quantum polynomial time algorithms respectively.
- *Vector/list indexed by a set.*
  Let S and V be finite sets. Let $\mathbf{v}$ be a vector/list with entries in V, indexed by S, i.e. $\mathbf{v}$ contains a value in V for each $s \in S$. We denote by $\mathbf{v}[s] \in V$ the value corresponding to s.
  For a subset $C \subseteq S$, we write $\mathbf{v}[C]$ for the vector obtained by restricting the indices of $\mathbf{v}$ to the set C. For example, if $\mathbf{v}'$ is a vector indexed by C, then, by $\mathbf{v}[C] = \mathbf{v}'$ we mean that $\mathbf{v}[s] = \mathbf{v}'[s]$ for all $s \in C$.
- *Asymptotic notation.*
  - *Big-O.* Let $f, g : \mathbb{N} \to \mathbb{R}$. We write $f \leq O(g)$ if $\exists c, n_0$ such that for all $n \geq n_0$, $f(n) \leq cg(n)$.
  - *Big-O on* Λ. Take $\Lambda \subseteq \mathbb{N}$ to be an infinite subset of $\mathbb{N}$. Then, by $f \leq O(g)$ on Λ, we mean that there are c, n_0 such that for all $n \geq n_0$ in Λ, $f(n) \leq cg(n)$.

  We define Big-Ω on Λ as a similar generalisation of the Big-Ω notation.

3.2 QFHE Scheme

We formally define the notion of Quantum Homomorphic Encryption that we will employ in the rest of the paper. Note that, as in [KLVY23] we only require security to hold against PPT provers.

Definition 5 (Quantum Homomorphic Encryption)**.** *A Quantum Homomorphic Encryption scheme for a class of circuits $\mathcal{C}$ is a tuple of algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *with the following syntax:*

- Gen *is PPT. It takes a unary input 1^n, and outputs a secret key* sk.
- Enc *is QPT. It takes as input a secret key* sk *and a quantum state $|\psi\rangle$, and outputs a ciphertext $|c\rangle$. We require that if $|\psi\rangle$ is classical, i.e. a standard basis state, then* Enc *becomes a PPT algorithm. In particular, $|c\rangle$ is also classical.*
- Eval *is QPT. It takes as input a tuple $(C, |\xi\rangle, |c\rangle)$, where*
  1. *$C : \mathcal{H} \times (\mathbb{C}_2)^{\otimes n} \to (\mathbb{C}_2)^{\otimes m}$ is a quantum circuit in $\mathcal{C}$,*
  2. *$|\xi\rangle \in \mathcal{H}$ is a quantum state,*
  3. *$|c\rangle$ is a ciphertext encrypting an n-qubit state.*

  Eval *computes a quantum circuit* $\mathsf{Eval}_C(|\xi\rangle, |c\rangle)$*, which outputs a ciphertext $|c_{\mathsf{out}}\rangle$. We require that, if C has classical output, then* Eval_C *also has classical output.*
- Dec *is QPT. It takes as input a secret key* sk *and a ciphertext $|c\rangle$, and outputs a state $|\phi\rangle$. If $|c\rangle$ is classical, then $|\phi\rangle$ is also classical.*

The correctness and security conditions are as follows:

1. ***Correctness:*** *For any quantum circuit* $C : \mathcal{H} \times (\mathbb{C}_2)^{\otimes n} \to (\mathbb{C}_2)^{\otimes m}$ *in* $\mathcal{C}$*, there is a negligible function* negl*, such that the following holds. For any quantum state* $|\xi\rangle \in \mathcal{H}$*, any* n*-qubit state* $|\psi\rangle$*, any* $\lambda \in \mathbb{N}$*, any* $\mathsf{sk} \leftarrow \mathsf{Gen}(1^\lambda)$*, and any* $|c\rangle \leftarrow \mathsf{Enc}_{\mathsf{sk}}(|\psi\rangle)$*, the following two states are* $\mathsf{negl}(\lambda)$*-close in trace distance.*
   - $C(|\xi\rangle \otimes |\psi\rangle)$,
   - $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Eval}_C(|\xi\rangle, |c\rangle))$.
2. ***Security:*** *For all PPT distinguishers* D*, for all polynomial functions* poly*, there exists a negligible function* negl *such that, for any two strings* $x_0, x_1 \in \{0,1\}^{\mathsf{poly}(\lambda)}$*, for all* λ*,*

$$\left| \Pr\left[D(|c_0\rangle) = 1 : \begin{array}{c} \mathsf{sk} \leftarrow \mathsf{Gen}(1^\lambda) \\ |c_0\rangle \leftarrow \mathsf{Enc}_{\mathsf{sk}}(x_0) \end{array} \right] - \Pr\left[D(|c_1\rangle) = 1 : \begin{array}{c} \mathsf{sk} \leftarrow \mathsf{Gen}(1^\lambda) \\ |c_1\rangle \leftarrow \mathsf{Enc}_{\mathsf{sk}}(x_1) \end{array} \right] \right| \leq \mathsf{negl}(\lambda) . \quad (4)$$

A Quantum *Fully* Homomorphic Encryption scheme (QFHE) is a Quantum Homomorphic Encryption scheme for the class of all poly-size quantum circuits.

Form of encryption. For the rest of this work, we restrict to (public key) QFHE schemes where encryption takes the following form. For an n-qubit state $|\psi\rangle$,

$$(U_k |\psi\rangle, \hat{k}) \leftarrow \mathsf{QFHE.Enc}_{\mathsf{sk}}(\psi) , \quad (5)$$

where $\{U_k\}_{k\in K}$ forms a group (potentially up to global phases) and $K(n)$ is a finite set. Furthermore, $\hat{k}$ is a classical fully homomorphic encryption of k, which can be decrypted using a PPT algorithm (specified by the QFHE scheme) given sk.

We show that such QFHE schemes also satisfy the following property.

Classical evaluation of classical circuits. There is a PPT procedure cEval that is the classical version of Eval. More precisely, we require that the following properties hold:

- For any classical ciphertext c produced by running Enc on a classical message m, and any classical circuit C, it holds that $c' \leftarrow \mathsf{cEval}(C, c)$ is such that $C(m) = \mathsf{Dec}(\mathsf{sk}, c)$.
- cEval and Eval produce identical distributions when run on identical classical inputs (i.e. ciphertexts and circuits).

Claim 6 (Classical Eval). Consider any (public-key) QFHE scheme whose ciphertext is of the form in Equation 5. Then, there exists a PPT procedure cEval, satisfying the classical evaluation of classical circuits property stated above.

All known QFHE schemes satisfying Definition 5 [Mah20, Bra18] are public key schemes that satisfy both properties listed above. In particular, the first property above (form of encryption) is satisfied with U_k being a Pauli pad and $k = (x, z) \in \{0,1\}^{2n}$ encoding which Pauli operator is applied, i.e. $U_k = X^x Z^z$. We end by outlining how the second property (classical evaluation of classical circuits) is proved generically as stated in Claim 6.

Proof sketch for Claim 6. We will show how the classical evaluation works by looking at homomorphic evaluation of the "xor" (addition) and "and" (multiplication) operations, that are universal for classical circuits. Consider two ciphertexts $\tilde{c}_0$ and $\tilde{c}_1$ corresponding to two messages m_0 and m_1, i.e. $\tilde{c}_0$ and $\tilde{c}_1$ are of the form $\tilde{c}_0 = (m_0 \oplus k_0, c_{k_0})$ and $\tilde{c}_1 = (m_1 \oplus k_1, c_{k_1})$ where $c_{k_0} \leftarrow \mathsf{FHE.Enc}_{\mathsf{pk}}(k_0)$ and $c_{k_1} \leftarrow \mathsf{FHE.Enc}_{\mathsf{pk}}(k_1)$. Then to homomorphically evaluate the addition, cEval will just add $\tilde{c}_0$ and $\tilde{c}_1$, namely compute $\tilde{c}_{m_0 \oplus m_1} = (m_0 \oplus k_0 \oplus m_1 \oplus k_1, c_{k_0} + c_{k_1}) = ((m_0 \oplus m_1) \oplus (k_0 \oplus k_1), c_{k_0 \oplus k_1})$, where $c_{k_0 \oplus k_1}$ is the FHE encryption of $k_0 \oplus k_1$.

For the homomorphic multiplication operation, we will first compute: $(m_0 \oplus k_0) \cdot (m_1 \oplus k_1) = (m_0 \cdot m_1) \oplus (k_0 \cdot m_1 \oplus k_1 \cdot m_0 \oplus k_0 \cdot k_1)$. As a result, to complete the homomorphic evaluation, it suffices to show how to compute a FHE encryption of the pad $k_0 \cdot m_1 \oplus k_1 \cdot m_0 \oplus k_0 \cdot k_1$. This can be done as follows. First, compute the FHE encryptions $c_{m_0 \oplus k_0} = \mathsf{FHE.Enc}_{\mathsf{pk}}(m_0 \oplus k_0)$ and $c_{m_1 \oplus k_1} = \mathsf{FHE.Enc}_{\mathsf{pk}}(m_1 \oplus k_1)$. Then given c_{k_0} and $c_{m_0 \oplus k_0}$, the algorithm will homomorphically compute the xor of the two messages (under encryption), resulting in obtaining $c_{m_0} = \mathsf{FHE.Enc}_{\mathsf{pk}}(m_0)$ (and analogously $c_{m_1} = \mathsf{FHE.Enc}_{\mathsf{pk}}(m_1)$). Next, given c_{k_1} and c_{m_0} one can homomorphically compute $c_{m_0 \oplus k_1} = \mathsf{FHE.Enc_{pk}}(m_0 \oplus k_1)$ and similarly, $c_{m_1 \oplus k_0} = \mathsf{FHE.Enc_{pk}}(m_1 \oplus k_0)$. The procedure will then sample a uniformly random r and compute $c_r = \mathsf{FHE.Enc_{pk}}(r)$.

Finally, by homomorphically adding all the pieces together we get the final ciphertext of the multiplication: $\tilde{c}_{m_0 \cdot m_1} := ((m_0 \cdot m_1) \oplus (k_0 \cdot m_1 \oplus k_1 \cdot m_0 \oplus k_0 \cdot k_1 \oplus r), \mathsf{FHE.Enc_{pk}}(k_0 \cdot m_1 \oplus k_1 \cdot m_0 \oplus k_0 \cdot k_1 \oplus r))$. It is clear to see that the ciphertext is in the right QFHE form, and moreover, due to the uniform pad r we also have that the corresponding key is uniform, hence, ensuring that the output of cEval is identically distributed to the output of Eval.

□

3.3 Noisy Trapdoor Claw-Free Functions

Our work requires trapdoor claw-free functions for two purposes:

- They are used to instantiate Mahadev's QFHE scheme [Mah20].
- They are used to construct the *oblivious pad*.

However, the only known constructions of TCFs that satisfy the properties needed to instantiate QFHE (e.g. the property of Definition 8) are "noisy" versions (NTCFs), and they are based on the hardness of LWE. We define NTCFs next. Before doing so, we point out that, while an NTCF is needed to instantiate QFHE based on current knowledge, any TCF (even based on quantum insecure assumptions, like Diffie-Hellman) suffices to build the *oblivious pad* (however, for simplicity, we still use NTCFs in Section 8).

For readers familiar with the area, note that the "adaptive hardcore bit" property is not required for any of the constructions in this work.

Definition 7 (NTCF family; paraphrased from [BCM+21].)**.** *Let λ be a security parameter. Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets and $\mathcal{K}$ be a finite set of keys (these sets implicitly depend on λ). A family of functions*

$$\mathcal{F} = \{f_{\mathsf{pk},b} : \mathcal{X} \to \mathcal{D}_{\mathcal{Y}}\}_{\mathsf{pk} \in \mathcal{K}, b \in \{0,1\}}$$

is called a noisy trapdoor claw free function (NTCF) family if the following conditions hold (for each λ):

1. ***Efficient Function Generation:*** *There exists an efficient probabilistic algorithm* Gen *that generates a (public) key in $\mathcal{K}$ together with a trapdoor (secret key)* sk*:*

$$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda).$$

2. ***Trapdoor Injective Pair:***

   (a) Trapdoor: *There exists an efficient deterministic algorithm* Inv *such that with overwhelming probability over the choice of* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$*, the following holds:*

$$\mathsf{Inv}(\mathsf{sk}, b, y) = x$$

   for all $b \in \{0,1\}, x \in \mathcal{X}$ and $y \in \mathsf{Supp}(f_{\mathsf{pk},b}(x))$.

   (b) Injective pair: *For all keys $\mathsf{pk} \in \mathcal{K}$, there exists a perfect matching $\mathcal{R}_{\mathsf{pk}} \subseteq \mathcal{X} \times \mathcal{X}$ such that $f_{\mathsf{pk},0}(x_0) = f_{\mathsf{pk},1}(x_1)$ if and only if $(x_0, x_1) \in \mathcal{R}_{\mathsf{pk}}$. Such a pair (x_0, x_1) is referred to as a "claw".*

3. ***Efficient Range Superposition.*** *For all keys $\mathsf{pk} \in \mathcal{K}$ and $b \in \{0,1\}$ there exists a function $f'_{\mathsf{pk},b} : \mathcal{X} \to \mathcal{D}_{\mathcal{Y}}$ such that the following holds:*

   (a) $\mathsf{Inv}(\mathsf{sk}, b, y) = x_b$ *and* $\mathsf{Inv}(\mathsf{sk}, b \oplus 1, y) = x_{b \oplus 1}$ *for all $(x_0, x_1) \in \mathcal{R}_{\mathsf{pk}}$ and $y \in \mathsf{Supp}(f'_{\mathsf{pk},b}(x_b))$.*

   (b) There is an efficient deterministic procedure Chk *s.t. for all $b \in \{0,1\}$, $\mathsf{pk} \in \mathcal{K}$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$,*

$$\mathsf{Chk}(\mathsf{pk}, b, x, y) = \begin{cases} 1 & \text{if } y \in \mathsf{Supp}(f'_{\mathsf{pk},b}(x)) \\ 0 & \text{else.} \end{cases}$$

   Observe that Chk *is not provided the secret trapdoor* sk.

(c) *For each* $\mathsf{pk} \in \mathcal{K}$ *and* $b \in \{0,1\}$*, it holds that*

$$\mathbb{E}_{x \leftarrow \mathcal{X}}[H^2(f_{\mathsf{pk},b}(x), f'_{\mathsf{pk},b}(x))] \leq \mathsf{negl}(\lambda) \tag{6}$$

for some negligible function negl*, where* H^2 *denotes the Hellinger distance.*
Further, there is an efficient procedure Samp *that on input* pk *and* $b \in \{0,1\}$ *prepares the state*

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \sqrt{(f'_{\mathsf{pk},b}(x))(y)}\, |x\rangle\, |y\rangle\,.$$

4. ***Claw-Free Property.*** *For any PPT adversary* $\mathcal{A}$*,*

$$\Pr\left[(x_0, x_1) \in \mathcal{R}_{\mathsf{pk}} : \begin{array}{c} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda) \\ (x_0, x_1) \leftarrow \mathcal{A}(\mathsf{pk}) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

When $\mathcal{F}$ *is used with other primitives, we refer to the algorithms* $(\mathsf{Gen}, \mathsf{Inv}, \mathsf{Samp}, \mathsf{Chk})$ *and the various sets* $(\mathcal{X}, \mathcal{D}_{\mathcal{Y}}, \mathcal{Y})$ *with* $\mathcal{F}$ *prefixed, e.g.* $\mathsf{Gen}(1^\lambda)$ *is referred to as* $\mathcal{F}.\mathsf{Gen}(1^\lambda)$*.*

We also implicitly require that the classical version of *efficient range superposition* holds: uniform elements from the set $\mathcal{X}$ (and therefore $\mathcal{D}_{\mathcal{Y}}$ and $\mathcal{Y}$) can be efficiently sampled classically. This is the case for most NTCFs.

We define an additional property of NTCFs, which we will invoke directly in the soundness analysis of our proof of quantumness in Section 5.2.

Definition 8 ("Hiding a bit in the xor")**.** *Let* $\mathcal{F} = \{f_{\mathsf{pk},b} : \mathcal{X} \to \mathcal{D}_{\mathcal{Y}}\}_{\mathsf{pk} \in \mathcal{K}, b \in \{0,1\}}$ *be an NTCF family (as in Definition 7). We say that* $\mathcal{F}$ *"hides a bit in the xor" if* Gen *takes an additional bit of input* s*:*

$$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, s)\,,$$

such that the following holds (in addition to the properties already satisfied by pk *and* sk *in Definition 7). For all* $s \in \{0,1\}$*, if* $(\mathsf{pk}, \mathsf{sk})$ *is in the support of* $\mathsf{Gen}(1^\lambda, s)$*, then* $s = x_0[1] \oplus x_1[1]$ *for all* $(x_0, x_1) \in \mathcal{R}_{\mathsf{pk}}$*, where* $x_0[1]$ *and* $x_1[1]$ *denote the first bits of* x_0 *and* x_1 *respectively, and* $\mathcal{R}_{\mathsf{pk}}$ *is as in Definition 7. Moreover, for all QPT algorithms* $\mathcal{A}$*, there exists a negligible function* negl *such that, for all* λ*,*

$$\Pr[s \leftarrow \mathcal{A}(\mathsf{pk}) : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{F}.\mathsf{Gen}(1^\lambda, s)\ ,\ s \leftarrow \{0,1\}] \leq 1/2 + \mathsf{negl}(\lambda)\,. \tag{7}$$

The following theorem is implicit in [Mah20].

Theorem 9 ([Mah20])**.** *There exists an NTCF family with the property of Definition 8, assuming the quantum hardness of the Learning With Errors (LWE) problem.*

Part I

Even Simpler Proofs of Quantumness

4 The Proof of Quantumness

For an informal description of our proof of quantumness, we refer the reader to Section 2.6. Here we provide a formal description. Our proof of quantumness makes use of a NTCF family $\mathcal{F} = \{f_{\mathsf{pk},b} : \mathcal{X} \to \mathcal{D}_{\mathcal{Y}}\}_{\mathsf{pk} \in \mathcal{K}, b \in \{0,1\}}$ (as in Definition 7) that additionally satisfies the property of Definition 8, i.e. a claw-free function pair hides a bit in the xor of the first bit of any claw.

We parse the domain $\mathcal{X}$ as $\mathcal{X} = \{0,1\} \times \mathcal{V}$. Then, a bit more precisely, the property of Definition 8 says the following. Let $s \in \{0,1\}$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{F}.\mathsf{Gen}(1^\lambda, s)$. Let (μ_0, v_0) and (μ_1, v_1) be such that $f_{\mathsf{pk},0}(\mu_0, v_0) = f_{\mathsf{pk},1}(\mu_1, v_1)$. Then $\mu_0 \oplus \mu_1 = s$.

Our proof of quantumness invokes the procedure $\mathcal{F}.\mathsf{Samp}$ from the "Efficient Range Superposition" property in Definition 7. For simplicity, we describe our proof of quantumness assuming that the "Efficient Range Superposition" property holds exactly, i.e. the RHS of Equation (6) is zero. The actual procedure $\mathcal{F}.\mathsf{Samp}$ is indistinguishable from the exact one, up to a negligible distinguishing advantage in the security parameter.

Construction 1 (Proof of Quantumness)

Let $\lambda \in \mathbb{N}$ be a security parameter.

- **Message 1**: The verifier samples $s \leftarrow \{0,1\}$. Then, she samples $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{F}.\mathsf{Gen}(1^\lambda, s)$. The verifier sends pk to the prover.

- **Message 2:** The prover uses $\mathcal{F}.\mathsf{Samp}$ to create the state

$$\frac{1}{\sqrt{2|\mathcal{X}|}} \sum_{b \in \{0,1\}, x \in \mathcal{X}, y \in \mathcal{Y}} \sqrt{f_{\mathsf{pk}}(x)(y)} \,|b\rangle\,|x\rangle\,|y\rangle \tag{8}$$

  This state can be created by preparing a qubit in the state $|+\rangle$ (and a sufficiently large auxiliary register), and then running the $\mathcal{F}.\mathsf{Samp}$ controlled on the first qubit.

  For $b \in \{0,1\}$, let (μ_b^y, v_b^y) be such that $y \in \mathsf{Supp}(\mu_b^y, v_b^y)$ (this is a unique element by the properties of the NTCF). Then, we can rewrite the state as:

$$\begin{aligned}
&\frac{1}{\sqrt{2|\mathcal{X}|}} \sum_{b \in \{0,1\}} \sqrt{f_{\mathsf{pk}}(\mu_0^y, v_0^y)(y)} \,|b\rangle\,|\mu_b^y, v_b^y\rangle\,|y\rangle \\
=&\frac{1}{\sqrt{2|\mathcal{X}|}} \sum_{b \in \{0,1\}} \sqrt{f_{\mathsf{pk}}(\mu_0^y, v_0^y)(y)} \,|b\rangle\,|\mu_0^y \oplus b \cdot s, v_b^y\rangle\,|y\rangle
\end{aligned}$$

  The prover applies Hadamard gates to every qubit in the "v_b^y" register, and then measures all qubits except the first. Let the output be (μ, d, y). Then, the resulting post-measurement state of the first qubit is (up to global phases):

$$\begin{cases} |0\rangle + (-1)^{d \cdot (v_0^y + v_1^y)}\,|1\rangle & \text{if } s = 0 \\ |\mu \oplus \mu_0^y\rangle & \text{if } s = 1 \end{cases} \tag{9}$$

  The prover returns (μ, d, y) to the verifier.

- **Message 3:** The verifier samples $c \leftarrow \{0,1\}$, and sends c to the prover.

- **Message 4:**
  - If $c = 0$, the prover measures the qubit in the basis

$$\{\cos(\frac{\pi}{8})\,|0\rangle + \sin(\frac{\pi}{8})\,|1\rangle\,, -\sin(\frac{\pi}{8})\,|0\rangle + \cos(\frac{\pi}{8})\,|1\rangle\}\,.$$

  - If $c = 1$, the prover measures in the basis

$$\{\cos(-\frac{\pi}{8})\,|0\rangle + \sin(-\frac{\pi}{8})\,|1\rangle\,, -\sin(-\frac{\pi}{8})\,|0\rangle + \cos(-\frac{\pi}{8})\,|1\rangle\}\,.$$

  The prover returns the outcome b to the verifier.

- **Verifier's final computation:**
  - If $s = 0$, the verifier runs $(\mu_0^y, v_0^y) \leftarrow \mathcal{F}.\mathsf{Inv}(\mathsf{sk}, 0, y)$ and $(\mu_1^y, v_1^y) \leftarrow \mathcal{F}.\mathsf{Inv}(\mathsf{sk}, 1, y)$. She sets $a = d \cdot (v_0^y \oplus v_1^y)$. Finally, she outputs accept if $a \oplus b = c$, and reject otherwise.
  - If $s = 1$, the verifier runs $(\mu_0^y, v_0^y) \leftarrow \mathcal{F}.\mathsf{Inv}(\mathsf{sk}, 0, y)$. She sets $a = \mu \oplus \mu_0^y$. Finally, she outputs accept if $a \oplus b = 0$, and reject otherwise.

5 Analysis

5.1 Correctness

Correctness as stated below is straightforward to verify.

Theorem 10 (Correctness)**.** *There exists a QPT algorithm $\mathcal{A}$ and a negligible function* negl *such that, for all λ,*

$$\Pr[\mathcal{A} \text{ wins in Algorithm 1}] \geq \cos^2(\pi/8) - \mathsf{negl}(\lambda).$$

Proof. The QPT algorithm $\mathcal{A}$ follows the steps of the prover in Algorithm 1. Then, correctness follows from the fact that the state in Equation (9) is one of the four BB84 states, and a straightforward calculation. One can also realize that the prover's measurement is Bob's ideal CHSH measurement corresponding to question c, and that the verifier is precisely checking the appropriate CHSH winning condition based on s. The negligible loss in the correctness probability comes from the fact that, as we mentioned earlier, the procedure $\mathcal{F}.\mathsf{Samp}$ actually generates a state that is only negligibly close to that of Equation (8). ☐

5.2 Soundness

Theorem 11. *For any PPT prover $\mathcal{A}$, there exists a negligible function* negl *such that, for all λ,*

$$\Pr[\mathcal{A} \text{ wins in Algorithm 1}] \leq \frac{3}{4} + \mathsf{negl}(\lambda)\,.$$

Proof. We show this by giving a reduction from a prover $\mathcal{A}$ for Algorithm 1 to an adversary $\mathcal{A}'$ breaking the property of Definition 8 satisfied by $\mathcal{F}$, i.e. predicting the bit s "hidden in the claw-free pair". Specifically, we show that if $\mathcal{A}$ wins with probability $\frac{3}{4} + \delta$ for some $\delta \geq 0$, then $\mathcal{A}'$ can guess the the bit s from Equation (7) with probability at least $\frac{1}{2} + 2\delta$. $\mathcal{A}'$ proceeds as follows, for security parameter λ:

(i) $\mathcal{A}'$ receives pk (where pk is sampled according to $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{F}.\mathsf{Gen}(1^\lambda, s)$ for some uniformly random s).

(ii) $\mathcal{A}'$ runs $\mathcal{A}$ on "first message" pk. Let z be the output.

(iii) Continue running $\mathcal{A}$ on "second message" $c = 0$. Let b_0 be the received output.

(iv) Rewind $\mathcal{A}$ to just before step (iii). Run $\mathcal{A}$ on "second message" $c = 1$. Let b_1 be the received output.

(v) Output the guess $s' = b_0 \oplus b_1$.

We show that $\Pr[\mathcal{A}' \text{ wins}] \geq \frac{1}{2} + 2\delta$. Using the notation introduced above, first notice that when b_0 is a response that a Verifier from Algorithm 1 would accept, we have, by construction, that

$$a \oplus b_0 = 0\,, \tag{10}$$

where a is defined as in Algorithm 1 (given $\mathcal{A}$'s first response z, and sk). We refer to such a b_0 as "valid". Similarly, if b_1 is valid, then, by construction,

$$a \oplus b_1 = s\,. \tag{11}$$

Summing Equations (10) and (11) together gives $b_0 \oplus b_1 = s$ (where s is the bit that was actually sampled in (i)). Since $\mathcal{A}'$ outputs precisely $s' = b_0 \oplus b_1$, we have that

$$\Pr[\mathcal{A}' \text{ wins}] = \Pr[s' = s] \geq \Pr[b_0 \text{ and } b_1 \text{ are valid}]\,. \tag{12}$$

Now, for fixed pk and z, define

$$p_{win,0}^{\mathsf{pk},z} := \Pr[b_0 \text{ is valid} \,|\, \mathsf{pk}, z]\,,$$

and define $p^{\mathsf{pk},z}_{win,1}$ analogously using b_1. Then, we have

$$\begin{aligned}
\Pr[\mathcal{A}' \text{ wins}] &\geq \Pr[b_0 \text{ and } b_1 \text{ are valid}] \\
&= \mathop{\mathbb{E}}_{\substack{s \leftarrow \{0,1\} \\ (\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{F}.\mathsf{Gen}(1^\lambda, s) \\ z \leftarrow \mathcal{A}(\mathsf{pk})}} \left[p^{\mathsf{pk},z}_{win,0} \cdot p^{\mathsf{pk},z}_{win,1} \right] \\
&\geq \mathop{\mathbb{E}}_{\substack{s \leftarrow \{0,1\} \\ (\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{F}.\mathsf{Gen}(1^\lambda, s) \\ z \leftarrow \mathcal{A}(\mathsf{pk})}} \left[p^{\mathsf{pk},z}_{win,0} + p^{\mathsf{pk},z}_{win,1} - 1 \right] && (13) \\
&= 2 \cdot \mathop{\mathbb{E}}_{\substack{s \leftarrow \{0,1\} \\ (\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{F}.\mathsf{Gen}(1^\lambda, s) \\ z \leftarrow \mathcal{A}(\mathsf{pk})}} \left[\frac{1}{2} \left(p^{\mathsf{pk},z}_{win,0} + p^{\mathsf{pk},z}_{win,1} \right) \right] - 1\,, && (14)
\end{aligned}$$

where (13) follows from the inequality $x \cdot y \geq x + y - 1$ for all $x, y \in [0,1]$ (subtracting x from both sides makes this inequality apparent).

Now, notice that

$$\mathop{\mathbb{E}}_{\substack{s \leftarrow \{0,1\} \\ (\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{F}.\mathsf{Gen}(1^\lambda, s) \\ z \leftarrow \mathcal{A}(\mathsf{pk})}} \left[\frac{1}{2} \left(p^{\mathsf{pk},z}_{win,0} + p^{\mathsf{pk},z}_{win,1} \right) \right] = \Pr[\mathcal{A} \text{ wins}]\,. \tag{15}$$

Thus, plugging this into (14), gives

$$\Pr[\mathcal{A}' \text{ wins}] \geq 2 \cdot \Pr[\mathcal{A} \text{ wins}] - 1 = 2 \cdot \left(\frac{3}{4} + \delta \right) - 1 = \frac{1}{2} + 2\delta\,, \tag{16}$$

as desired.

□

Part II

A Computational Test of Contextuality—for size 2 contexts

6 Contextuality Games

We start by defining contextuality games. This section uses the list/vector notation from Section 3.1.

Definition 12 (Contextuality game, strategy, value)**.** *A contextuality game* G *is specified by a tuple* $(Q, A, C^{\text{all}}, \text{pred}, \mathcal{D})$ *where*

- Q *is a set of questions.*
- A *is a set of answers.*
- C^{all} *is a set of subsets of* Q*. We refer to an element of* C^{all} *as a* context.
- $\mathcal{D}$ *is a distribution over contexts in* C^{all}*, and*
- pred *is a binary-valued function. It takes as input pairs of the form* $(\mathbf{a}, C)$*, where* $C \in C^{\text{all}}$*, and* $\mathbf{a} \in A^{|C|}$*. We think of* $\mathbf{a}$ *as being indexed by the questions in* C*, and we write* $\mathbf{a}[q] \in A$ *to represent the answer to question* $q \in C$*.*

G *can be thought of as a 2-message game between a referee and a player that proceeds as follows:*

(i) The referee samples a context $C \leftarrow \mathcal{D}$ *and sends* C *to the prover.*

(ii) The player responds with $\mathbf{a} \in A^{|C|}$*.*

(iii) The referee accepts if $\text{pred}(\mathbf{a}, C) = 1$*.*

A strategy *for the game* G *is specified by a family* P *of probability distributions (one for each* $C \in C^{\text{all}}$*), where* $P[\mathbf{a}|C]$ *can be thought of as the probability of answering* $\mathbf{a}$ *on questions in* C*.*

We define the value of G *with respect to a strategy* P *to be*

$$\text{val}(P) := \sum_{C \in C^{\text{all}}, \mathbf{a} \in A^{|C|}} \text{pred}(\mathbf{a}, C) \cdot P(\mathbf{a}|C) \cdot \Pr_{\mathcal{D}}[C].$$

where $\Pr_{\mathcal{D}}[C]$ *denotes the probability assigned to* C *by* $\mathcal{D}$*.*

Remark 13. One can assume that all contexts $C \in C^{\text{all}}$ have the same size without loss of generality by adding additional observables, and having the predicate pred remain unchanged (i.e. the predicate does not depend on the values taken by the additional observables).

Let us consider the magic square as a contextuality game to clarify the notation.

Example 4 (Magic Square Contextuality Game)**.** The set of questions is

$$Q := \{(1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)\},$$

$A := \{+1, -1\}$ and C^{all} consists of all rows and columns of this matrix, i.e. subsets of the form $\{(r,1), (r,2), (r,3)\}$ for $r \in \{1,2,3\}$ and $\{(1,c), (2,c), (3,c)\}$ for $c \in \{1,2,3\}$. The distribution $\mathcal{D}$ uniformly selects an element $C \leftarrow C^{\text{all}}$, i.e. it samples either the row or column uniformly. The predicate $\text{pred}(\mathbf{a}, C)$ is 0 unless the following holds:

$$\prod_{q \in C} \mathbf{a}[q] = \begin{cases} -1 & \text{if } C = \{(1,3), (2,3), (3,3)\} \\ 1 & \text{else.} \end{cases}$$

Call this game, $\mathsf{G}_{\text{masq}} := (Q, A, C^{\text{all}}, \text{pred}, \mathcal{D})$, the magic square game.

The game only becomes meaningful when the strategies that the players follow are somehow restricted. Thus, to consider non-contextual and quantum values associated with a contextuality game G, we must first define the corresponding strategies.

For the following two definitions, let $\mathsf{G} = (Q, A, C^{\text{all}})$ be a contextuality game.

Definition 14 (Classical/Non-Contextual Strategy for G | $\mathcal{S}_{NC}(\mathsf{G})$)**.** *We write a joint probability distribution over answers to* all *questions in Q as*

$$P_{\text{joint}}[\mathbf{a_{joint}}] \in [0, 1]$$

where $\mathbf{a_{joint}}$ is a vector of length $|Q|$, indexed by Q. A strategy P is a non-contextual strategy *if $P[\mathbf{a}|C]$ is derived from some joint probability $P_{\text{joint}}[\mathbf{a_{joint}}]$, i.e.*

$$P(\mathbf{a}|C) = \sum_{\mathbf{a_{joint}}:\mathbf{a_{joint}}[C]=\mathbf{a}} P_{\text{joint}}(\mathbf{a_{joint}})$$

where by $\mathbf{a_{joint}}[C] = \mathbf{a}$ we mean that $\mathbf{a_{joint}}[\tilde{q}] = \mathbf{a}[\tilde{q}]$ for all $\tilde{q} \in C$.[18] *The set of all non-contextual strategies is denoted by $\mathcal{S}_{NC}(\mathsf{G})$.*

We now define a quantum strategy for G. It would be helpful to first define "qstrat"—a state and collection of observables—consistent with a contextuality game G.

The quantum strategy for a contextuality game G is defined as follows.

Definition 15 (Quantum Strategy for G | $\mathcal{S}_{Qu}(\mathsf{G})$)**.** *Given a contextuality game* G*, consider a triple*

$$\mathsf{qstrat} := (\mathcal{H}, |\psi\rangle, \mathbf{O}),$$

where $\mathcal{H}$ is a Hilbert space, $|\psi\rangle \in \mathcal{H}$ is a quantum state and $\mathbf{O}$ is a list of Hermitian operators (observables), indexed by the questions Q (i.e. $\mathbf{O}[q]$ is an observable for each $q \in Q$) acting on $\mathcal{H}$. Furthermore, suppose that $\mathbf{O}$ satisfies the following:

(i) $\mathsf{Spectr}[\mathbf{O}[q]] \subseteq A$ *for all $q \in Q$, i.e. the values returned by the observables correspond to potential answers in the game.*

(ii) $[\mathbf{O}[q], \mathbf{O}[q']] = 0$ for all $q, q' \in C$, for each $C \in C^{\text{all}}$, i.e. operators corresponding to the same context are compatible.

A strategy is a quantum strategy*, i.e. $P \in \mathcal{S}_{Qu}(G)$, iff there is such a triple* $\mathsf{qstrat} := (\mathcal{H}, |\psi\rangle, \mathbf{O})$ *satisfying*

$$P[\mathbf{a}|C] = \Pr[\textit{Start with } |\psi\rangle \textit{ measure } \mathbf{O}[C] \textit{ and obtain } \mathbf{a}] \quad \forall\, \mathbf{a}, C. \tag{17}$$

To be concrete, consider the Magic square as an example.

Example 5 (qstrat for the magic square)**.** For the magic square as in Example 1, the optimal quantum strategy corresponds to using $\mathsf{qstrat} := (\mathcal{H}, |\psi\rangle, \mathbf{O})$ where $\mathcal{H} := \mathbb{C}^2 \times \mathbb{C}^2$, $|\psi\rangle \in \mathcal{H}$ is any fixed state, say $|\psi\rangle = |00\rangle$, and $\mathbf{O}$ is specified by the following (indexed by the questions Q in some canonical way):

$$\begin{array}{ccc} \mathbb{I} \otimes X & Z \otimes \mathbb{I} & Z \otimes X \\ X \otimes \mathbb{I} & \mathbb{I} \otimes Z & X \otimes Z \\ X \otimes X & Z \otimes Z & XZ \otimes XZ. \end{array} \tag{18}$$

Note that this is not the most general strategy, i.e. one could have different operators for each context and the operators in a given context may not even commute. Why then, do we restrict to the strategies defined above? This is because, as explained briefly in the introduction, the appeal of a contextuality test is that even by only measuring commuting observables at any given time, one can find scenarios where the idea of pre-determined values of observables becomes untenable.

While conceptually appealing, the main obstacle to testing contextuality using the contextuality game is that it is unclear how the provers' strategies can be restricted to the ones above. As noted earlier, when one considers a Bell game as a contextuality game, spatial separation automatically restricts the provers' strategies in this way.

Assuming that the provers follow only these restricted strategies, one can nevertheless define the classical/non-contextual and quantum value of the contextuality game.

[18] Note that $\mathbf{a_{joint}}$ is a vector of length $|Q|$ while $\mathbf{a}$ is a vector of length $|C|$.

Definition 16 (Non-Contextual and Quantum Value of a Contextuality Game G)**.** *Let* $\mathsf{G} = (Q, A, C^{\text{all}}, \text{pred}, \mathcal{D})$ *be a contextuality game (see Definition 12 and recall the definition of* val*). The non-contextual value of* G *is given by*

$$\text{valNC} := \max_{P \in \mathcal{S}_{NC}(\mathsf{G})} \text{val}(P) \quad \textit{and similarly} \quad \text{valQu} := \max_{P \in \mathcal{S}_{Qu}(\mathsf{G})} \text{val}(P).$$

is the quantum value of G*.*

Before moving further, we quickly note what these values are for the magic square.

Example 6 (Magic Square Contextuality Game (cont.))**.** Let G_{masq} be as in Example 4 and note that

$$\text{valNC} = 5/6, \quad \text{while} \quad \text{valQu} = 1$$

using the Magic-Square con-instance as in Example 5.

The KCBS example (Example 3) mentioned in the introduction, in this notation, can be expressed as follows.

Theorem 17 (KCBS)**.** *There exists a contextuality game* G *with* $\text{valNC} = 0.8 < \text{valQu} = \frac{2}{\sqrt{5}} \approx 0.8944$ *where the quantum strategy can be realised using qutrits and five binary valued observables (i.e.* $\mathsf{qstrat} := (\mathcal{H}, |\psi\rangle, \mathbf{O})$ *is such that* $\dim \mathcal{H} = 3$*,* $|Q| = |\mathbf{O}| = 5$ *and* $A = \{\pm 1\}$*).*

7 Criteria for being an operational test of contextuality

We briefly mentioned in the technical overview that an "operational test of contextuality" has a close correspondance with a contextuality game. In this section, we formalise what we mean by this correspondence.

As alluded to in the discussion above and in Section 2.2, it is reasonable to assert that the most general non-contextual physically relevant model of computation is simply a probabilistic poly time Turing machine. Thus, any proof of quantumness, i.e. any test that distinguishes a PPT machine from a quantum machine, may be taken to be a proof of contextuality. In other words, quantumness and contextuality become equivalent notions, in accordance with this definition.

This, however, is unsatisfactory because quantumness could be arising from some other non-classical feature of quantum mechanics which may have no a priori connection to contextuality. For instance, (assuming factoring is hard for a PPT machine) equivalence of quantumness and contextuality would mean that factoring serves as a proof of contextuality. Yet, it is unclear how Shor's algorithm demonstrates contextuality in any direct way.

One proposal could be that a satisfactory test must be "universal" in the sense that corresponding to each contextuality game, there should be a systematic way to construct a corresponding test such that the completeness and soundness values of the test correspond to the quantum and noncontextual value of the underlying contextuality game. A little thought shows that this is not enough, as illustrated by the example below.

Example 7. The example uses a general non-interactive proof of quantumness as an ingredient: Let $\mathsf{PoQ} = (\mathsf{Gen}, \mathsf{Verify}, \mathsf{Cert})$ denote a proof of quantumness protocol where $\mathsf{Gen}, \mathsf{Verify}$ are PPT algorithms and Cert is a QPT algorithm. Here, Gen generates a challenge, Cert generates a response to the challenge and Verify tests whether the response is valid. For simplicity, we take them to satisfy the property that no PPT machine can make Verify accept while Cert always produces a valid certificate, i.e. PoQ has perfect completeness and soundness. Given a contextuality game G, the protocol is the following: the verifier runs a proof of quantumness protocol PoQ and if the prover passes, the verifier accepts with probability $\mathsf{G}.\text{valQu}$ and if the prover fails, accepts with probability $\mathsf{G}.\text{valNC}$.

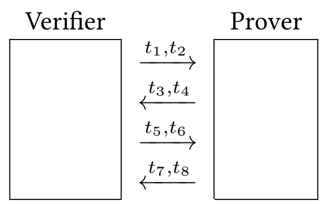
Verifier		Prover
$\mathsf{chall} \leftarrow \mathsf{PoQ.Gen}$	$\xrightarrow{\mathsf{chall}}$	
	$\xleftarrow{\pi}$	
if π is a valid response for chall (i.e. $\mathsf{PoQ.Verify}(\mathsf{chall}, \pi) = 1$)		
accept with probability $\mathsf{G}.\text{valNC}$.		
Otherwise		
accept with probability $\mathsf{G}.\text{valQu}$.		

Evidently, this protocol has completeness $c = \mathsf{G}.\mathrm{valQu}$ and soundness $s = \mathsf{G}.\mathrm{valNC}$.

Example 7 is unsatisfactory as a test of contextuality because, even though its soundness and completeness values correspond to the classical and quantum values of the game G, this correspondence is artifical: the prover is not asked a single question that is related to G. We now formalise a stronger and more natural notion of correspondence. To this end, we introduce some notation.

7.1 Notation

As a starting point towards a candidate "operational test" $\mathcal{P}$ for G, consider a proof of quantumness protocol $\mathcal{P}$ involving a PPT verifier V and a prover P, with soundness s and completeness c. To be concrete, suppose protocol $\mathcal{P}$ involves four messages. Further suppose that the messages are parsed as follows:



For instance t_1 could be a public key and t_2 could be an encrypted message.

Mapping messages in $\mathcal{P}$ to questions/answers in G. Let $\mathsf{G} = (Q, A, C^{\mathrm{all}}, \mathrm{pred}, \mathcal{D})$. We require that $\mathcal{P}$ classifies each message into one of three categories: (1) a question in G, (2) an answer in G or (3) other. More precisely, we require that $\mathcal{P}$ specifies a map map_i for each message t_i.
For indices i corresponding to messages *received by the prover* (in our example, t_1, t_2, t_5, t_6), map_i is either

- the constant $\perp$ output map, $\mathsf{map}^{\perp}$ (i.e. $\mathsf{map}^{\perp}$ outputs $\perp$ on all inputs) or
- a map $r_i : C_i \to Q$ from the set of possible messages C_i to a question in the game G.

One can think of r_i as a "decoding map" for the underlying question.
For indices i corresponding to messages *sent by the prover* (in our example t_3, t_4, t_7, t_8), map_i is either

- the constant $\perp$ output map $\mathsf{map}^{\perp}$ or
- a map $s_i : A \to C_i$ from the set of answers A in the game to possible messages C_i.

Similarly, one can think of s_i as an "encoding map" for the underlying answer. Continuing in the same vein as the example above, suppose $t_1 = \mathsf{pk}$ is a public key for some encryption scheme and $t_2 \leftarrow \mathsf{Enc}_{\mathsf{pk}}(q)$ is an encryption of a question $q \in Q$. Then, a natural choice for the corresponding maps is $\mathsf{map}_1 = \mathsf{map}^{\perp}$ and $\mathsf{map}_2 = \mathsf{Dec}_{\mathsf{sk}}$ where sk is the secret key corresponding to pk. Note that we gave one choice for these maps but $\mathcal{P}$ can specify these maps arbitrarily. The non-triviality arises from the requirements we place on $\mathcal{P}$, using these maps.

Faithfulness to G. In order to capture the fact that protocol $\mathcal{P}$ is faithfully executing an instance of the game G, and not rewarding some other capability of the prover, we consider two families of "simulators". These are computationally unbounded machines and are meant to simulate either a classical or a quantum strategy. The idea is that using $\{\mathsf{map}_i\}_i$ one can isolate the messages that correspond to questions and answers in the game G. One can then define machines that answer these questions non-contextually, or using a quantum strategy, and can behave arbitrarily otherwise. More precisely, we have the following:

- A classical simulator $S_{\tau,\mathsf{aux}}$ is an unbounded machine that is designed to interact with V and responds to "encoded" questions (as specified by $\{\mathsf{map}_i\}_i$) using some non-contextual assignment $\tau : Q \to A$ i.e. the simulator decodes the message t_i using r_i to recover a question $q \in Q$, computes $a = \tau(q)$ and responds with the encoding $s_j(a)$, for each pair of message indices (i, j) corresponding to a question and its answer. It responds to the remaining messages using an arbitrary strategy specified by[19] aux. We denote by $\mathbf{S}_{\mathrm{NC}} = \{S_{\tau,\mathsf{aux}}\}_{\tau,\mathsf{aux}}$ the set of all classical simulators.

[19]Here, aux may be thought of as describing the "program of a Turing Machine" and may involve potentially unbounded computation.

- A quantum simulator $S_{\mathsf{qstrat},\mathsf{aux}}$ is an unbounded machine that is also designed to interact with V and responds to the "encoded" questions using $\mathsf{qstrat} = (|\psi\rangle, \mathbf{O})$ of G (see Definition 15) but responds to the remaining messages using an arbitrary strategy specified by aux. We denote by $\mathbf{S}_{\mathsf{qu}} = \{S_{\mathsf{qstrat},\mathsf{aux}}\}_{\mathsf{qstrat},\mathsf{aux}}$ the set of all quantum simulators.

In addition to the simulators, we will also require $\mathcal{P}$ to specify an efficient classical procedure qProver that maps any (classical description of a) quantum strategy

$$\mathsf{qstrat} = (|\psi\rangle, \underbrace{\{O_1, O_2 \dots\}}_{=\mathbf{O}})$$

for G to a QPT prover

$$\mathsf{qProver}(\mathsf{qstrat}) = P_{\mathsf{qstrat}}$$

that is designed to interact with the verifier V in protocol $\mathcal{P}$. qProver must satisfy the following *"marginal" requirement*: the behaviour of P_{qstrat} when asked questions within a context $\{q_1 \dots q_k\} = C \in C^{\mathrm{all}}$ under the encoding specified by $\{\mathsf{map}_i\}$, depends only on the state $|\psi\rangle$ and observables $\mathbf{O}[C] = \{O_{q_1}, \dots O_{q_k}\}$ but not on the remaining observables $\mathbf{O}[Q \backslash C] = \mathbf{O} \backslash \mathbf{O}[C] = \mathbf{O} \backslash \{O_{q_1}, \dots O_{q_k}\}$, specified by qstrat. Just as the case when qstrat is played in G and questions are asked in C, the remaining observables don't make any difference.

7.2 Criteria

We can now state our definition. We ignore negligible additive factors for clarity.

Definition 18 (Operational Test of Contextuality)**.** *We say $\mathcal{P}$ is an* operational test of contextuality *if there exist s and c (where $s < c$) such that, in addition to being a proof of quantumness with soundness s and completeness c, $\mathcal{P}$ is* faithful to *some* contextuality game G. *We say $\mathcal{P}$ is* faithful to $\mathsf{G} = (Q, A, C^{\mathrm{all}}, \mathsf{pred}, \mathcal{D})$ *(with parameters s and c) if the following hold:*

1. *Well-formedness: $\mathcal{P}$ must specify maps $\{\mathsf{map}_i\}_i$ and the procedure* qProver *as described above. Moreover, for any possible set Q' of questions that the verifier V asks in a single execution, i.e.*

$$Q' := \{\mathsf{map}_i(t_i)\}_{i:t_i = r_i}, \tag{19}$$

   we require that the questions belong to some context, i.e. $Q' \subseteq C$ for some context $C \in C^{\mathrm{all}}$.

2. G*-soundness: For all classical simulators, i.e. $S \in \mathbf{S}_{\mathrm{NC}}$, the probability that the verifier V accepts when interacting with S should be at most the classical value s, i.e.*

$$\Pr(1 \leftarrow \langle V, S \rangle) \leq s \quad \forall \quad S \in \mathbf{S}_{\mathsf{NC}}.$$

3. *Decision Faithfulness: For all $S_{\tau,\mathsf{aux}} \in \mathbf{S}_{\mathsf{NC}}$, consider the questions Q' asked by V (see Equation (19)) in an execution of $\langle V, S_{\tau,\mathsf{aux}} \rangle$.*
   (a) *If Q' asks all questions in some context, i.e. $Q' = C$ for some $C \in C^{\mathrm{all}}$, then the verifier outputs* $\mathsf{pred}(\tau[C], C)$.
   (b) *Otherwise, the verifier outputs* 1.

4. G*-completeness: Given* qProver*, the following must hold:*
   (a) Quantum completeness.
   First, there should exist $\mathsf{qstrat} = (|\psi\rangle, \{O_1, O_2 \dots\})$ *(as in Definition 15) such that* $\Pr[1 \leftarrow \langle V, P_{\mathsf{qstrat}} \rangle] = c$ *where* $\mathsf{qProver}(\mathsf{qstrat}) =: P_{\mathsf{qstrat}}$.
   Second, for every P_{qstrat}, there is a quantum simulator, i.e. $S_{\mathsf{qstrat},\mathsf{aux}} \in \mathbf{S}_{\mathsf{qu}}$, satisfying the following: the transcript $\mathsf{transcript}(V, S_{\mathsf{qstrat},\mathsf{aux}})$ *produced by the interaction of $S_{\mathsf{qstrat},\mathsf{aux}}$ with V is distributed identically to the transcript* $\mathsf{transcript}(V, P_{\mathsf{qstrat}})$ *produced by the interaction of P_{qstrat} with V. In particular, this implies*

$$\exists\ S_{\mathsf{qstrat},\mathsf{aux}} \in \mathbf{S}_{\mathsf{qu}}, \ \textit{s.t.} \ \Pr[1 \leftarrow \langle V, S_{\mathsf{qstrat},\mathsf{aux}} \rangle] = \Pr[1 \leftarrow \langle V, P_{\mathsf{qstrat}} \rangle].$$

   (b) Classical completeness.
   For every qstrat *consisting of commuting observables, i.e. $[O_i, O_j] = 0$ for all $O_i, O_j \in \mathbf{O}$, there is a PPT prover P such that* $\mathsf{transcript}(V, P)$ *is distributed identically to* $\mathsf{transcript}(V, P_{\mathsf{qstrat}})$.

7.3 Justification

According to Definition 18, if a proof of quantumness protocol $\mathcal{P}$ is faithful to a contextuality game G, it must specify a mapping $\{\mathsf{map}_i\}_i$ relating it to G. Now if a prover wins with probability greater than s, from Condition 2 we know that there is no way of interpreting the behaviour of this prover as being consistent with a non-contextual assignment in G (via $\{\mathsf{map}_i\}_i$). Condition 3, ensures that the criterion for rewarding/penalising a prover is determined solely by the predicate of G (via $\{\mathsf{map}_i\}_i$). Finally, Condition 4 requires that there is a systematic procedure $\mathsf{qProver}$ for converting any quantum strategy in G to a prover in $\mathcal{P}$. The marginal condition on $\mathsf{qProver}$ basically ensures that the procedure does not artificially produce a different prover when all observables commute vs when they do not. Condition (a) says that $\mathsf{qProver}$ produces provers whose behaviour is consistent with a quantum simulator implementing the same quantum strategy and that the best strategy achieves the completeness value c. Condition 4 (b) ensures that when observables do commute (i.e. we are in the non-contextual setting), the behaviour of the prover produced by $\mathsf{qProver}$ can be understood in terms of a non-contextual model, i.e. a PPT machine.

We now introduce these requirements sequentially, illustrating why each one of them plays a crucial role in ruling out unsatisfactory notions of operational tests of contextuality.

- Condition 1 is a very basic requirement. For instance, in Example 7, one could assign map $\mathsf{map}^{\perp}$ to every message in the protocol and this would satisfy 1.

- Condition 2 ensures the following: Suppose that the messages that map to $\perp$ are answered by using unbounded computational power. Even in this case, as long as the messages that correspond to questions/answers in G (as specified by the protocol $\mathcal{P}$) are answered non-contextually, the condition requires that one cannot do better than a PPT machine.
  For instance, Example 7 fails to satisfy this requirement: an unbounded classical simulator $S \in \mathbf{S}_{\mathsf{NC}}$ can easily produce a valid proof of quantumness and make the verifier accept with probability $\mathsf{G}.\mathrm{valQu}$ which is greater than the soundness value $s = \mathsf{G}.\mathrm{valNC}$. Clearly, Example 7 was a very simple construction. Consider the less trivial construction in Example 8 below where the verifier asks questions in the game G, in addition to requesting a proof of quantumness, but only considers the prover's answers when the PoQ π is valid. Using natural maps $\{\mathsf{map}_i\}$ (i.e. $\mathsf{map}_1 = \mathsf{map}_2 = \mathsf{map}_4 = \mathsf{map}_5 = \mathbb{I}, \mathsf{map}_3 = \mathsf{map}_6 = \mathsf{map}^{\perp}$) it is immediate that Condition 2 is satisfied by this construction: even if a simulator $S \in \mathbf{S}_{\mathsf{NC}}$ provides a valid proof of quantumness, the verifier does not accept with probability more than $\mathsf{G}.\mathrm{valNC}$. Yet, this construction is intuitively unsatisfactory as a test of contextuality because it is completely neglecting the answers a_1, a_2 given by a classical prover.

Example 8. Given a contextuality game (with size-two contexts) G, the verifier proceeds as described (where chall and π are as in Example 7). This protocol satisfies Conditions 1 and 2, but fails to satisfy 3 (a) (see Definition 18).

Verifier		Prover
$(q_1, q_2) = C \leftarrow C^{\mathrm{all}}, \mathsf{chall} \leftarrow \mathsf{PoQ.Gen}$	$\xrightarrow{(t_1,t_2,t_3)=(q_1,q_2,\mathsf{chall})}$	
If π is invalid, i.e. $\mathsf{PoQ.Verify}(\mathsf{chall}, \pi) = 0$, accept with probability $\mathsf{G}.\mathrm{valNC}$ else, i.e. $\mathsf{PoQ.Verify}(\mathsf{chall}, \pi) = 1$ output $\mathsf{G.pred}(a_1, a_2, q_1, q_2)$	$\xleftarrow{(t_4,t_5,t_6)=(a_1,a_2,\pi)}$	

- Condition 4 is also satisfied by Example 8 and this illustrates why the last condition (below) is so crucial.

- Condition 3 is where Example 8 finally fails: Consider two simulators $S_{\tau,\mathsf{aux}}, S_{\tau,\mathsf{aux}'} \in \mathbf{S}_{\mathsf{NC}}$ where the assignment τ corresponds to a game value $v < \mathsf{G}.\mathrm{valNC}$ and aux corresponds to producing the correct proof of quantumness while aux' corresponds to producing an invalid proof of quantumness. Condition 3 (a) requires that $\Pr[1 \leftarrow \langle V, S_{\tau,\mathsf{aux}}\rangle] = \Pr[1 \leftarrow \langle V, S_{\tau,\mathsf{aux}'}\rangle]$ but, $\Pr[1 \leftarrow \langle V, S_{\tau,\mathsf{aux}}\rangle] = v$ and $\Pr[1 \leftarrow \langle V, S_{\tau,\mathsf{aux}'}\rangle] = \mathsf{G}.\mathrm{valNC}$.

- The relevance of Condition 3 (b) is evident from Example 9 below, where the verifier starts by asking for a PoQ certificate and only if this is valid does it ask the questions for the contextuality game G; otherwise it simply rejects. Intuitively, this is unsatisfactory because a classical prover is not even able to see the questions in G, let alone answer them (unlike Example 8). Yet, none of the previous conditions are violated. Such cases are excluded by Condition 3 (b) because it requires that no prover can be penalised unless all questions in some context are asked.

Example 9. Given a contextuality game (with size-two contexts) G, the verifier proceeds as described below (where chall, π are as in Example 7). It is easy to see that that this protocol satisfies Conditions 1 and 2 but not 3 (b), with $s = \mathsf{G}.\mathrm{valNC}$ and $q = \mathsf{G}.\mathrm{valQu}$.

Verifier		Prover
$\mathsf{chall} \leftarrow \mathsf{PoQ.Gen}$	$\xrightarrow{t_1=\mathsf{chall}}$	
	$\xleftarrow{t_2=\pi}$	
If π is valid (i.e. $\mathsf{PoQ.Verify}(\mathsf{chall}, \pi) = 1$) ask $(q_1, q_2) = C \leftarrow C^{\mathrm{all}}$, receive (a_1, a_2), and output $\mathsf{pred}(a_1, a_2, q_1, q_2)$.	If π valid: $\xrightarrow{(t_3,t_4)=(q_1,q_2)}$ $\xleftarrow{(t_5,t_6)=(a_1,a_2)}$ $\xrightarrow{\mathsf{out}=\mathsf{pred}(a_1,a_2,q_1,q_2)}$	
Else reject.	otherwise: $\xrightarrow{\mathsf{out}=\mathsf{reject}}$	

Here, we take the soundness value[20] $s = \mathsf{G}.\mathrm{valNC}$ while the quantum value is $q = \mathsf{G}.\mathrm{valQu}$.

- All examples we have considered so far, have failed at least one of the criteria discussed above. However, it is easy to verify that Example 8 and Example 9 above both satisfy Condition 4 (a), using the natural choice for $\mathsf{qProver}$. We now consider an example, Example 10 (below), that satisfies all conditions above Conditions 1, 2, 3 and Condition 4 (a) and yet, it is unsatisfactory.
  - The idea in Example 10 is to "help" the prover recover the post-measurement state, only if it produces a proof of quantumness certificate. More precisely, the verifier asks a QFHE encrypted question $\hat{q}_1$ together with a proof of quantumness challenge chall. The prover is expected to respond with an encrypted answer $\hat{a}_1$ together with a certificate for the proof of quantumness, π. The verifier then sends the second question q_2 together with a string s. If the prover provided a valid certificate π, the string s is a key that allows the prover to decrypt any homomorphic computation performed using $\hat{q}_1$ (in particular, the prover can learn the post-measurement state; and also q_1 itself). Otherwise s is some irrelevant random string. The verifier expects an answer a_2 and accepts if $\mathsf{pred}(a_1, a_2, q_1, q_2) = 1$.
  - This seems unsatisfactory because the construction hinges entirely on the proof of quantumness: a prover who is given the ability to pass the proof of quantumness (as a black-box), but who is otherwise classical, is able to win the game with probability 1 (assuming $\mathsf{QFHE.Eval}$ is classical for classical circuits).
  - Formally, it is easy to check that all conditions are satisfied (using natural maps) except Condition 4 (b). However, there does not seem to be any procedure $\mathsf{qProver}$ that satisfies both quantum completeness and classical completeness simultaneously. For instance, the natural choice for $\mathsf{qProver}(\mathsf{qstrat}) = P_{\mathsf{qstrat}}$ (recall $\mathsf{qstrat} = (\{O_1, O_2 \dots\}, |\psi\rangle)$ is simply to answer $\hat{q}_1$ by measuring O_{q_1} on $|\psi\rangle$ homomorphically, produce π corresponding to chall, then use the key sent by the verifier to recover the post-measurement state and measure O_{q_2} on this state to obtain the answer a_2. Now, it is immediate that even when all observables commute, there is no PPT prover that can produce the same transcript as P_{qstrat} (when interacting with the verifier)—a PPT prover cannot produce a valid proof of quantumness certificate. One might try to modify the $\mathsf{qProver}$ procedure so that it does not produce a proof of quantumness certificate π when the observables commute, but the marginal requirement rules out any such attempt. Finally, if $\mathsf{qProver}$ never produces a valid proof of quantumness, it cannot produce a P_{qstrat} that makes the verifier accept with probability $c > s$.

Example 10. Given a contextuality game (with size-two contexts) G, the verifier proceeds as described below (where chall, π are as in Example 7 while the QFHE scheme is as in Definition 5).

[20] Even though it is clear that every PPT prover succeeds with at most negligible probability; If one, in fact, takes $s = 0$, then Condition 3 (a) is already violated.

Verifier		Prover
$(q_1, q_2) = C \leftarrow C^{\text{all}}$		
$\mathsf{sk}_0, \mathsf{sk}_1 \leftarrow \mathsf{QFHE.Gen}$		
$\hat{q}_1 \leftarrow \mathsf{QFHE.Enc}_{\mathsf{sk}_1}(q_1)$		
$\mathsf{chall} \leftarrow \mathsf{PoQ.Gen}$		
	$\xrightarrow{(t_1,t_2)=(\hat{q}_1,\mathsf{chall})}$	
	$\xleftarrow{(t_3,t_4)=(\hat{a}_1,\pi)}$	
$b = \mathsf{PoQ.Verify}(\mathsf{chall}, \pi)$		
$a_1 = \mathsf{QFHE.Dec}_{\mathsf{sk}_1}(\hat{a}_1)$		
	$\xrightarrow{(t_5,t_6)=(q_2,\mathsf{sk}_b)}$	
	$\xleftarrow{t_7=a_2}$	
output $\mathsf{pred}(a_1, a_2, q_1, q_2)$		

We conclude this discussion by noting that in Condition 4b, having computational indistinguishability of the transcripts does not suffice. To see this, observe that in Example 10, if messages t_2, t_4, t_5, t_6, t_7 are exchanged under homomorphic encryption (with an independent key), then it is straightforward to construct a PPT machine that produces a transcript that is computationally indistinguishable from that produced by P_{qstrat}. In fact, one can apply such a transformation quite generically, rendering Condition 4b irrelevant, if only computational indistinguishability is demanded.

8 OPad | Oblivious U-Pad

Before describing our compiler, we introduce the oblivious pad primitive, and show how to construct it.

8.1 Definition

Let $\mathbf{U} := \{U_k\}_{k\in K}$ be a set of unitaries acting on a Hilbert space $\mathcal{H}$, where K is a finite set.

Definition 19. *An Oblivious* $\mathbf{U}$*-Pad (or an OPad) is a tuple of algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *as follows:*

- Gen *is a PPT algorithm with the following syntax:*
  - *Input:* 1^λ *(a security parameter in unary).*
  - *Output:* $(\mathsf{pk}, \mathsf{sk})$.
- Enc *is A QPT algorithm with the following syntax:*
  - *Input:* pk*, a state* ρ *on* $\mathcal{H}$*.*
  - *Output: a state* σ *(also on* $\mathcal{H}$*) and a string* s*.*
- Dec *is a classical polynomial-time deterministic algorithm with the following syntax:*
  - *Input:* sk, s*.*
  - *Output:* $k \in K$*.*
- Samp *is a PPT algorithm with the following syntax:*
  - *Input:* pk
  - *Output: a string* s

We require the following.

- *Correctness: There exists a negligible function* negl *such that, for all* ρ *on* $\mathcal{H}$*,* $\lambda \in \mathbb{N}$*, the following holds with probability at least* $1 - \mathsf{negl}(\lambda)$ *over sampling* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$*, and* $(\sigma, s) \leftarrow \mathsf{Enc}(\mathsf{pk}, \rho)$*:*

$$\sigma \approx_{\mathsf{negl}(\lambda)} U_k \rho U_k^\dagger \,,$$

  where $k = \mathsf{Dec}(\mathsf{sk}, s)$*.*

- *Soundness: For any PPT prover P, there exists a negligible function* negl *(not necessarily equal to the previous one) such that, for all $\lambda \in \mathbb{N}$, P wins in the following game with probability at most $1/2 + \mathsf{negl}(\lambda)$.*

Prover		*Challenger*
		$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$
	$\xleftarrow{\mathsf{pk}}$	
(The PPT prover can proceed arbitrarily)		
	$\xrightarrow{s}$	
		$b \leftarrow \{0,1\}$ $k_0 \leftarrow K$ $k_1 = \mathsf{Dec}(\mathsf{sk}, s)$
	$\xleftarrow{k_b}$	
	$\xrightarrow{b'}$	
		Accept if $b' = b$

- *Classical range sampling:* Samp *can sample a string s from the same distribution as* Enc*, i.e. for all $\rho \in \mathcal{H}$, and for all unbounded distinguishers D, it holds that*

$$|\Pr[1 \leftarrow D(s) : s \leftarrow \mathsf{Samp}(\mathsf{pk})] - \Pr[1 \leftarrow D(s') : (s', \sigma) \leftarrow \mathsf{Enc}(\mathsf{pk}, \rho)]| \leq \mathsf{negl}(\lambda)$$

where $\mathsf{pk} \leftarrow \mathsf{Gen}(1^\lambda)$.

8.2 Relation between the OPad and proofs of quantumness

On a first read, one may skip to Section 8.3 where we instantiate the OPad in the random oracle model. Here, we briefly discuss the relation between the OPad and proofs of quantumness. We describe how the OPad implies a proof of quantumness and discuss how proofs of quantumness can be used to realise some properties of the OPad.[21]

OPad implies proof of quantumness. A 4-message proof of quantumness is immediate. Consider the case where $\mathbf{U} = (\mathbb{I}, \sigma_x, \sigma_y, \sigma_z)$ indexed by $K = (0, x, y, z)$ of size 4. Observe that the security game for the OPad immediately yields a proof of quantumness. From the security guarantee, there is no PPT algorithm that wins with probability non-negligibly greater than $1/2$. Yet, there is a quantum prover that can distinguish k_0 from k_1 with probability at least $3/4$.

[21]The latter was pointed out by an anonymous QCrypt reviewer.

Prover		Challenger
	$\xleftarrow{\mathsf{pk}}$	$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$
Start with $\left|1\right\rangle$, and apply Enc on $\left|1\right\rangle$ to obtain $(U_k \left|1\right\rangle, s) \leftarrow \mathsf{Enc}(\mathsf{pk}, \left|1\right\rangle)$		
	$\xrightarrow{s}$	
		$b \leftarrow \{0,1\}$ $k_0 \leftarrow K$ $k_1 = \mathsf{Dec}(\mathsf{pk}, \mathsf{sk}, s)$
	$\xleftarrow{k_b}$	
Measure $U_{k_b}^\dagger U_k \left|1\right\rangle$ in the standard basis. If the outcome is $\left|1\right\rangle$, set $b' = 1$. Else set $b' = 0$.		
	$\xrightarrow{b'}$	accept if $b' = b$

It is elementary to check that the prover succeeds with probability $3/4$: when $b = 1$, the prover always reports $b' = 1$ but when $b = 0$ the prover reports $b' = 0$ with probability at least half (because for any $B \in \mathrm{U}$ there are two distinct $A, A' \in \mathbf{U}$ such that $\left\langle 0\right| AB \left|1\right\rangle = \left\langle 0\right| A'B \left|1\right\rangle = 1$).

One can also construct a 2-message proof of quantumness protocol where the challenger samples $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$, sends pk and accepts if the prover returns $k' = \mathsf{Dec}(\mathsf{sk}, s)$.

Non-interactive proofs of quantumness imply a weakened variant of OPad. Any non-interactive proof of quantumness protocol allows one to construct a weakened variant of OPad.[22] The idea is best illustrated by considering factoring. Let $\mathsf{OPad} =: (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and proceed as follows:

- Gen: Generates a random large composite number N and returns $(\mathsf{pk}, \mathsf{sk}) := (N, N)$.
- $\mathsf{Enc}(\mathsf{pk}, \left|\psi\right\rangle)$:
  - Samples $k \leftarrow K$,
  - Obtains a factor (using Shor's algorithm) F of N and defines $s := (k, F)$
  - Returns $(U_k \left|\psi\right\rangle, s)$
- $\mathsf{Dec}(\mathsf{sk}, s)$:
  - Parses $s = (k, F)$
  - If F is a factor of N then
    outputs k.
    Otherwise
    outputs $k' \leftarrow K$.

While correctness is immediate, this construction may seem unsound since Enc is revealing k in the clear. However, the point is that no PPT machine can produce a "valid encryption" (i.e. one that encodes a factor), and the decrypt procedure outputs a random value from K upon being given an "invalid encryption". Effectively, this means that a PPT machine can only have the decrypt algorithm output a random value from K which is exactly what the soundness condition demands.

There are two issues with this construction. First, the Dec procedure is randomised while the definition requires it to be deterministic. But more importantly, second, there is no Samp procedure that can sample from the range of Enc—no PPT procedure can produce correct factors of N.

[22]This observation is due to an anonymous QCrypt referee.

The first issue is not too hard to resolve (described below) but to resolve both simultaneously, we use the random oracle model.

Hardness of integer factoring implies OPad without efficient range sampling. We outline the idea here. Let $\mathcal{S} = (\mathcal{S}.\mathsf{Gen}, \mathcal{S}.\mathsf{Enc}, \mathcal{S}.\mathsf{Dec})$ denote a classically CCA-secure public key encryption such that (a) $\mathcal{S}.\mathsf{Dec}$ is a deterministic algorithm and (b) there is an efficient quantum algorithm QBreak that correctly decrypts any ciphertext without needing the secret key. Such a scheme is known, based on integer factoring [HKS13]. Suppose $(\mathsf{pk}', \mathsf{sk}') \leftarrow \mathcal{S}.\mathsf{Gen}(1^\lambda)$. Then the OPad is constructed as follows:

- $\mathsf{OPad}.\mathsf{Gen}(1^\lambda)$
  - Runs $(\mathsf{pk}', \mathsf{sk}') \leftarrow \mathcal{S}.\mathsf{Gen}(1^\lambda)$
  - Samples $\mathsf{pad} \leftarrow K$
  - Computes $\mathsf{pk} := (\mathsf{pk}', \mathcal{S}.\mathsf{Enc}_{\mathsf{pk}'}(\mathsf{pad}))$ and defines $\mathsf{sk} := (\mathsf{sk}', \mathsf{pad})$
  - Returns $(\mathsf{pk}, \mathsf{sk})$
- $\mathsf{OPad}.\mathsf{Enc}(\mathsf{pk}, |\psi\rangle)$
  - Samples $\tilde{k} \leftarrow K$
  - Recovers pad from pk using QBreak
  - Defines $k = \mathsf{pad} \oplus \tilde{k}$
  - Applies U_k to the input state $|\psi\rangle$
  - Computes $s \leftarrow \mathcal{S}.\mathsf{Enc}_{\mathsf{pk}'}(k)$
  - Returns $(U_k |\psi\rangle, s)$
- $\mathsf{OPad}.\mathsf{Dec}(\mathsf{sk}, s)$ returns $k := \mathcal{S}.\mathsf{Dec}_{\mathsf{sk}'}(s) \oplus \mathsf{pad}$.

Correctness is again immediate. Soundness also holds intuitively: no PPT prover should be able to win the security game for the OPad as long as $\mathcal{S}$ is CCA secure. However, again, we do not have efficient range sampling in this case.

8.3 Instantiating the OPad | Oblivious Pauli Pad

Algorithm 2 below shows how to instantiate an oblivious $\mathbf{U}$-pad in the random oracle model, where $\mathbf{U} = \{X^x Z^z\}_{x,z}$ where x, z are strings. We refer to this as an *oblivious Pauli pad*. The instantiation leverages ideas from the *proof of quantumness* protocol in [BKVV20]. For convenience, we restate the informal description from Section 2.5. We describe the encryption procedure for a single qubit, and using a TCF pair f_0, f_1 (rather than an NTCF as in Algorithm 2). To encrypt a multi-qubit state, one simply applies the following encryption procedure to each qubit.

We take $\mathsf{pk} = (f_0, f_1)$, and sk to be the corresponding trapdoor. Then, $\mathsf{Enc}_{\mathsf{pk}}$ is as follows:

(i) On input a qubit state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$, evaluate f_0 and f_1 in superposition, controlled on the first qubit, and measure the output register. This results in some outcome y, and the leftover state $\alpha |0\rangle |x_0\rangle + \beta |1\rangle |x_1\rangle$, where $f(x_0) = f(x_1) = y$.

(ii) Compute the random oracle "in the phase", to obtain $(-1)^{H(x_0)}\alpha |0\rangle |x_0\rangle + (-1)^{H(x_1)}\beta |1\rangle |x_1\rangle$. Measure the second register in the Hadamard basis. This results in a string d, and the leftover qubit state

$$|\psi_Z\rangle = Z^{d\cdot(x_0 \oplus x_1) + H(x_0) + H(x_1)} |\psi\rangle \ .$$

(iii) Repeat steps (i) and (ii) on $|\psi_Z\rangle$, but *in the Hadamard basis*! This results in strings y' and d', as well as a leftover qubit state

$$|\psi_{XZ}\rangle = X^{d'\cdot(x_0' \oplus x_1') + H(x_0') + H(x_1')} Z^{d\cdot(x_0 \oplus x_1) + H(x_0) + H(x_1)} |\psi\rangle \ ,$$

where x_0' and x_1' are the pre-images of y'.

Notice that the leftover qubit state $|\psi_{XZ}\rangle$ is of the form $X^x Z^z |\psi\rangle$ where x, z have the following two properties: (a) a verifier in possession of the TCF trapdoor can learn z and x given respectively y, d and y', d', and (b) no PPT prover can produce strings y, d as well as predict the corresponding bit z with non-negligible advantage (and similarly for x). Intuitively, this holds because a PPT prover that can predict z with non-negligible advantage must be querying the random oracle at *both* x_0 and x_1 with non-negligible probability. By simulating the random oracle (by lazy sampling, for instance), one can thus extract a claw x_0, x_1 with non-negligible probability, breaking the claw-free property.

Algorithm 2 Oblivious Pauli Pad

Let

- $\mathcal{F}$ be an NTCF family (see Definition 7).
- $|\psi\rangle \in \mathcal{H}$ be a state acting on J qubits.
- $H : \{0,1\}^* \to \{0,1\}$ denote the random oracle.

Define:

- $\mathsf{Gen}(1^\lambda)$:
  - Execute $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{F}.\mathsf{Gen}(1^\lambda)$ and output $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Enc}(\mathsf{pk}, \rho)$:
  - For each $j \in \{1 \dots J\}$,
    execute $\mathsf{qubitEnc}$ (as described in Algorithm 3) on qubit j of ρ using the public key pk, and use (k_j, s_j) to denote (k', s') as in Equation (24).
  - Let $k = (k_1 \dots k_J)$, and $s = (s_1 \dots s_J)$. Denote the resulting state by σ_k.
  - Return σ_k and s.
- $\mathsf{Dec}(\mathsf{sk}, s)$:
  - For each $j \in \{1 \dots J\}$
    denote by k_j the output of $\mathsf{qubitDec}(\mathsf{sk}, s_j)$ (as described in Algorithm 4).[23]
  - Return $k = (k_1 \dots k_j)$.
- $\mathsf{Samp}(\mathsf{pk})$:
  - For each $j \in \{1 \dots J\}$,
    sample $s_j \leftarrow \mathsf{sSamp}(\mathsf{pk})$ (as described in Algorithm 5).
  - Return $s = (s_1 \dots s_J)$.

[23]Note that qubitDec does not output a qubit, but just a classical string! Nonetheless, we choose to call it qubitDec, since the output are the quantum one-time pad keys associated to the qubitEnc procedure.

Algorithm 3 $\mathsf{qubitEnc}$

$\mathsf{qubitEnc}(\mathsf{pk}, |\psi\rangle)$ where $|\psi\rangle$ is a single qubit state (extended to mixed states by linearity)

- Without loss of generality, suppose the state of the qubit is given by $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.
- Proceed as follows

$$
\begin{aligned}
&|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \\
&\mapsto \sum_{x\in\mathcal{X},y\in\mathcal{Y}} \alpha\sqrt{(\mathcal{F}.f'_{\mathsf{pk},0}(x))(y)}\,|0\rangle|x\rangle|y\rangle + \beta\sqrt{(\mathcal{F}.f'_{\mathsf{pk},1}(x))(y)}\,|1\rangle|x\rangle|y\rangle && \text{Using } \mathcal{F}.\mathsf{Samp} \\
&\approx_\epsilon \sum_{x\in\mathcal{X},y\in\mathcal{Y}} \alpha\sqrt{(\mathcal{F}.f_{\mathsf{pk},0}(x))(y)}\,|0\rangle|x\rangle|y\rangle + \beta\sqrt{(\mathcal{F}.f_{\mathsf{pk},1}(x))(y)}\,|1\rangle|x\rangle|y\rangle && \text{where } \epsilon \text{ is negligible} \\
&\mapsto (\alpha|0\rangle|x_0\rangle + \beta|1\rangle|x_1\rangle)\;|y\rangle && \text{Measuring the last register to get some } y, \\
& && \text{where } y \text{ is s.t.} \\
& && \mathcal{F}.f_{\mathsf{pk},0}(x_0) = \mathcal{F}.f_{\mathsf{pk},1}(x_1) = y. \\
&\mapsto \left((-1)^{H(x_0)}\alpha|0\rangle|x_0\rangle + (-1)^{H(x_1)}\beta|1\rangle|x_1\rangle\right)\;|y\rangle && \text{Applying the random oracle} \\
& && H \text{ in the phase.} \\
&\mapsto \left((-1)^{d\cdot x_0 + H(x_0)}\alpha|0\rangle + (-1)^{d\cdot x_1 + H(x_1)}\beta|1\rangle\right)\;|d\rangle|y\rangle && \text{Measuring the second register} \\
& && \text{in the Hadamard basis to get some } d. \\
&= \underbrace{Z^{d\cdot(x_0\oplus x_1)+H(x_0)+H(x_1)}|\psi\rangle}_{=:|\phi\rangle}|d\rangle|y\rangle && \text{Up to a global phase.}
\end{aligned} \tag{20}
$$

- Relabel (d, y, x_0, x_1) to $(d_Z, y_Z, x_{Z,0}, x_{Z,1})$.
- Denote by $|\phi\rangle = \alpha'|+\rangle + \beta'|-\rangle$ the state of the first qubit in Equation (20). Continue as follows

$$
\alpha'|+\rangle + \beta'|-\rangle \tag{21}
$$

$$
\mapsto \sum_{x\in\mathcal{X},y\in\mathcal{Y}} \alpha\sqrt{(\mathcal{F}.f'_{\mathsf{pk},0}(x))(y)}\,|+\rangle|x\rangle|y\rangle + \beta\sqrt{(\mathcal{F}.f'_{\mathsf{pk},1}(x))(y)}\,|-\rangle|x\rangle|y\rangle \quad \text{By applying Samp.}
$$

$\vdots$

proceed as above to obtain

$\vdots$

$$
= X^{d\cdot(x_0\oplus x_1)+H(x_0)+H(x_1)}|\phi\rangle|d\rangle|y\rangle \quad \text{Up to a global phase.} \tag{22}
$$

- Relabel (d, y, x_0, x_1) to $(d_X, y_X, x_{X,0}, x_{X,1})$.
- Now note that the final state in Equation (22) can be written as

$$
X^{\mathsf{phase}(d_X, x_{X,0}, x_{X,1})} Z^{\mathsf{phase}(d_Z, x_{Z,0}, x_{Z,1})}|\psi\rangle \tag{23}
$$

  where $\mathsf{phase}(d, x_0, x_1) := d\cdot(x_0\oplus x_1) + H(x_0) + H(x_1)$.

- Define

$$
k' := (\mathsf{phase}(d_X, x_{X,0}, x_{X,1}), \mathsf{phase}(d_Y, x_{Y,0}, x_{Y,1})), \text{ and } s' := (d_X, y_X,\ d_Z, y_Z). \tag{24}
$$

- Return the state in Equation (23), and s'.

Algorithm 4 qubitDec

$\mathsf{qubitDec}(\mathsf{sk}, \underbrace{(d_X, y_X, d_Z, y_Z)}_{s'})$

- From y_X, compute $x_{X,0} = \mathcal{F}.\mathsf{Inv}(\mathsf{sk}, 0, y_X)$ and $x_{X,1} = \mathcal{F}.\mathsf{Inv}(\mathsf{sk}, 1, y_X)$.
- Similarly, from y_Z, compute $x_{Z,0} = \mathcal{F}.\mathsf{Inv}(\mathsf{sk}, 0, y_Z), x_{Z,1} = \mathcal{F}.\mathsf{Inv}(\mathsf{sk}, 1, y_Z)$.
- Compute k' as in Equation (24), i.e.

$$k' = (\mathsf{phase}(d_X, x_{X,0}, x_{X,1}), \mathsf{phase}(d_Y, x_{Y,0}, x_{Y,1})) \tag{25}$$

  where $\mathsf{phase}(d, x_0, x_1) := d \cdot (x_0 \oplus x_1) + H(x_0) + H(x_1)$

- Return k'.

Algorithm 5 sSamp

$\mathsf{sSamp}(\mathsf{pk})$

- Sample $x_X \leftarrow \mathcal{X}$ and $x_Y \leftarrow \mathcal{X}$ (where $\mathcal{X}$ is specified by the NTCF $\mathcal{F}$ and the security parameter).
- Evaluate $y_X = \mathcal{F}.f_{\mathsf{pk},0}(x_X)$ and $y_Y = \mathcal{F}.f_{\mathsf{pk},0}(x_Y)$.
- Sample $d_X \leftarrow \mathcal{X} \setminus \{0\}$ and $d_Y \leftarrow \mathcal{X} \setminus \{0\}$.
- Return (d_X, y_X, d_Z, y_Z).

Lemma 20 (Correctness and Efficient Range Sampling)**.** *Suppose $\mathcal{F}$ is an NTCF as in Definition 7. Then, Algorithm 2 satisfies both the correctness and the efficienc range sampling requirements in Definition 19.*

Proof. This is straightforward to verify given the properties of the NTCF. □

Lemma 21 (Soundness)**.** *Suppose $\mathcal{F}$ is an NTCF as in Definition 7. Then, Algorithm 2 satisfies the soundness requirement in Definition 19.*

Proof. We first analyse the simpler game $\mathcal{G}$ (described below) and then observe that the reasoning carries over to the soundness of Algorithm 2.

Prover ($\mathcal{P}$)		Challenger ($\mathcal{C}$)
		$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$
	$\xleftarrow{\mathsf{pk}}$	
	$\xrightarrow{y}$	
		$b \leftarrow \{0,1\}$ $h_0 \leftarrow \{0,1\}$ $h_1 = H(x_0) \oplus H(x_1)$ where $x_0 = \mathcal{F}.\mathsf{Inv}(\mathsf{sk}, 0, y)$ and $x_1 = \mathcal{F}.\mathsf{Inv}(\mathsf{sk}, 1, y)$.
	$\xleftarrow{h_b}$	
	$\xrightarrow{b'}$	
		Accept if $b' = b$

Intuitively, it is clear that no PPT prover $\mathcal{P}$ wins with probability more than $1/2 + \mathrm{negl}$ because if a PPT prover can distinguish h_0 from h_1, it must know H at both x_0 and x_1 with non-negligible probability. By simulating the random oracle, one can then construct a PPT algorithm to extract preimages x_0, x_1 of y. This violates the claw-free property of $\mathcal{F}$.

Formally, suppose that $\mathcal{P}$ succeeds with probability at most $1/2+\eta$ for some non-negligible function η. We show below that the following straightforward reduction extracts preimages x_0, x_1 of y with non-negligible probability:

Prover ($\mathcal{P}$), queries to H are simulated by $\mathcal{R}$		Reduction $\mathcal{R}$ that uses $\mathcal{P}$		Challenger $\mathcal{F}.\mathcal{C}$, for the claw-free property of $\mathcal{F}$
		Answer queries to H made by $\mathcal{P}$ by maintaining a database D. If the query to H is at x, check if D contains x, otherwise sample $\mathrm{val} \leftarrow \{0,1\}$, respond with val and append (x, val) to D.		$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$
	$\xleftarrow{\mathsf{pk}}$		$\xleftarrow{\mathsf{pk}}$	
	$\xrightarrow{y}$			
		$h_0 \leftarrow \{0,1\}$		
	$\xleftarrow{h_0}$			
	$\xrightarrow{b'}$			
		Ignore b'. Check if D has two inputs (x_0, x_1) with the same output y under $\mathcal{F}$ (i.e. $\mathcal{F}.\mathsf{Chk}(\mathsf{pk}, 0, x_0, y) = \mathcal{F}.\mathsf{Chk}(\mathsf{pk}, 1, x_1, y) = 1$). Otherwise, set $x_0 = x_1 = 0$.		
			$\xrightarrow{(x_0, x_1, y)}$	

We lower bound the success probability of $\mathcal{R}$. Assume, without loss of generality, that $\mathcal{P}$ does not repeat queries. Let E be the following event, during the execution of $\mathcal{P}$, interacting with $\mathcal{C}$: $\mathcal{P}$ makes a query at $x \in \{x_0, x_1\}$ such that after this query, both x_0 and x_1 have been queried.

Note that $\neg E$ means that, by an information theoretic argument, $\mathcal{P}$ cannot distinguish the random variable $H(x_0) \oplus H(x_1)$ from a uniformly random bit.

Suppose $\mathcal{P}$ interacts with $\mathcal{C}$. Then,

$$\Pr[\mathcal{P} \text{ wins}] = \Pr[\neg E] \Pr[\mathcal{P} \text{ wins}|\neg E] + \Pr[E] \Pr[\mathcal{P} \text{ wins}|E].$$

NB: $\Pr[b = 0|\neg E] = \Pr[b = 1|\neg E] = \frac{1}{2}$ because $\Pr[b = 0|\neg E] = \Pr[\neg E|b = 0] \Pr[b = 0] / \Pr[\neg E]$ and $\Pr[\neg E|b = 0] = \Pr[\neg E|b = 1]$ because until E happens, $b = 0$ and $b = 1$ cannot make any difference in the execution of the protocol.

Continuing,

$$\begin{aligned}\Pr[\mathcal{P} \text{ wins}] = \Pr[\neg E] \left(\Pr[b' = 0|b = 0 \wedge \neg E] \cancelto{1/2}{\Pr[b = 0|\neg E]} + \underbrace{\Pr[b' = 1|b = 1 \wedge \neg E]}_{=\Pr[b'=1|b=0 \wedge \neg E]} \cancelto{1/2}{\Pr[b = 1|\neg E]} \right) + \\ \Pr[E] \Pr[\mathcal{P} \text{ wins}|E] \\ = \Pr[\neg E] \cdot \frac{1}{2} + \Pr[E] \cdot \Pr[\mathcal{P} \text{ wins}|E] \\ = \frac{1}{2} + \Pr[E] \cdot \left(\Pr[\mathcal{P} \text{ wins}|E] - \frac{1}{2} \right)\end{aligned}$$

and since $\Pr[\mathcal{P} \text{ wins}] \geq \frac{1}{2} + \eta$, it implies that $\Pr[E] \geq \eta'$ for some other non-negligible function η'. Since $\Pr[\mathcal{R} \text{ wins}] = \Pr[E]$, we conclude $\mathcal{R}$, a PPT algorithm, wins against $\mathcal{F}.\mathcal{C}$ with probability η which contradicts the claw-free property of $\mathcal{F}$.

To use this result, we first need to generalise to the case of multiple ys. This is also quite straightforward. Consider the following game $\mathcal{G}'$:

Prover ($\mathcal{P}$)		Challenger ($\mathcal{C}$)
		$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$
	$\xleftarrow{\mathsf{pk}}$	
	$\xrightarrow{y_1 \cdots y_{2J}}$	
		$b \leftarrow \{0,1\}$ $h_0 \leftarrow \{0,1\}^{2J}$ $h_1 = h_{11}, h_{21} \ldots h_{2J,1}$ where $h_{j1} = H(x_{j0}) \oplus H(x_{j1})$ where in turn $x_{j,0} = \mathcal{F}.\mathsf{Inv}(\mathsf{sk}, 0, y_j)$ and $x_{j,1} = \mathcal{F}.\mathsf{Inv}(\mathsf{sk}, 1, y_j)$.
	$\xleftarrow{h_b}$	
	$\xrightarrow{b'}$	
		Accept if $b' = b$

Also consider the modified reduction $\mathcal{R}'$ which is the same as the reduction $\mathcal{R}$, except that it receives $y_1 \ldots y_{2J}$ and at the last step, where it checks D for any two inputs corresponding to any one of the $y_1 \ldots y_{2J}$. The success probability of $\mathcal{R}'$ can be lower bounded as before, except that the event E now becomes the following: $\mathcal{P}$ makes a query $x \in \{x_{j,0}, x_{j,1}\}_{j \in [2J]}$ such that all for at least some j, both $x_{j,0}$ and $x_{j,1}$ have been queried.

The last step is to relate the game $\mathcal{G}'$ with the soundness game in Definition 19. The main difference between the two is that instead of h_0 and h_1, the soundness game uses k_0 and k_1 where, for Dec given by the oblivious pauli pad (i.e. Algorithm 2), k_1 has the form (for j odd)

$$k_{j1} = (d_j \cdot (x_{j0} \oplus x_{j1}) + H(x_{j,0}) + H(x_{j,1}),\ d_{j+1} \cdot (x_{j+1,0} \oplus x_{j+1,1}) + H(x_{j+1,0}) + H(x_{j+1,1})).$$

The reasoning in computing the probability of E goes through as above. The reduction $\mathcal{R}'$ remains unchanged (it anyway was not computing k_1). Assuming that the claw-free property of $\mathcal{F}$ holds, we conclude that Algorithm 2 satisfies the soundness condition. □

In Section 9, we show how to use the oblivious pad, along with a QFHE scheme, to obtain a contextuality compiler. In order for this to be possible, the oblivious pad needs to be "compatible" with the QFHE scheme in the following sense.

Definition 22 (OPad compatible with QFHE)**.** *Suppose a* QFHE *scheme satisfies Definition 5 with the form of encryption of n-qubit states specified by $\{U_k\}_{k\in K}$ as in Equation (5). Then, an oblivious $\mathbf{U}$-Pad (or an* OPad*) as in Definition 19 is compatible with the* QFHE *scheme if $\mathbf{U} = \{U_k\}_{k\in K}$.*

While for our compiler, taking $\mathbf{U}$ to be the Pauli group suffices, we give a more general construction below.

8.4 Instantiating the OPad | General Oblivious U-Pad

We informally describe how one can instantiate the oblivious $\mathbf{U}$-Pad more generically for essentially any group of unitaries $\mathbf{U}$.

To describe the idea, for simplicity, let $\mathbf{V} = \{\mathbb{I}, X\}$ and suppose the input state is a single qubit state $|\psi\rangle$. In the Pauli Pad construction, we used TCFs to apply the unitary $V_k := X^{d\cdot(x_0\oplus x_1)+H(x_0)+H(x_1)} \in \mathbf{V}$ to $|\psi\rangle$ where $k = d \cdot (x_0 \oplus x_1) + H(x_0) + H(x_1)$. We had defined $s := (d, y)$ where y is the image of x_0 and x_1. The Enc procedure returned $(V_k |\psi\rangle, s)$. It was, however, not immediate how one would extend this construction to unitaries beyond the Pauli group.

Now, in order to apply a general unitary $U_k \in \mathbf{U}$ (which does not necessarily have to be a Pauli matrix), we use TCFs to generate a pair (k, s) as above but on auxiliary qubits, and then apply the unitary U_k on the input state $|\psi\rangle$. Intuitively, since a PPT algorithm producing (k, s) entails that it must know claws for y (as it must have evaluated $H(x_0)$ and $H(x_1)$), we conclude that this construction satisfies the soundness condition for an OPad.

More precisely, suppose the unitary group $\mathbf{U} = \{U_k\}_k$ is indexed by J-bit strings k and each unitary acts on a Hilbert space $\mathcal{H}$. The corresponding OPad may be instantiated as in Algorithm 6.

Algorithm 6 Oblivious $\mathbf{U}$-Pad for a general $\mathbf{U}$.

Let

- $\mathcal{F}$ be an NTCF family (see Definition 7),
- $\epsilon > 0$ be a fixed small constant, and
- $|\psi\rangle \in \mathcal{H}$
- $H : \{0,1\}^* \to \{0,1\}$ denote the random oracle.

Define:

- $\mathsf{Gen}(1^\lambda)$:
  - Execute $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{F}.\mathsf{Gen}(1^\lambda)$ and output $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Enc}(\mathsf{pk}, \rho)$:
  - For each $j \in \{1 \dots J\}$, execute $\mathsf{qSamp}_j(\mathsf{pk})$ (as described in Algorithm 7) to obtain $k = (k_1 \dots k_J)$ and $s = (s_1 \dots s_J)$.
  - Return $\sigma_k := U_k \rho U_k^\dagger$ along with s.
- $\mathsf{Dec}(\mathsf{sk}, s)$:
  - For each $j \in \{1 \dots J\}$,
    * Parse $s_j =: (d, y)$.
    * From y, compute $x_0 = \mathsf{Inv}(\mathsf{sk}, 0, y)$ and $x_1 = \mathsf{Inv}(\mathsf{sk}, 1, y)$.
    * Compute
    $$k_j(s_j) = \mathsf{phase}(d, x_0, x_1)$$
    where $\mathsf{phase}(d, x_0, x_1) := d \cdot (x_0 \oplus x_1) + H(x_0) + H(x_1)$ as before.
  - Return $k = (k_1 \dots k_j)$.
- $\mathsf{Samp}(\mathsf{pk})$:
  - For each $j \in \{1 \dots J\}$,
    sample $s_j \leftarrow \mathsf{sSamp}(\mathsf{pk})$ (as described in Algorithm 8).
  - Return $s = (s_1 \dots s_J)$.

Algorithm 7

$\mathsf{qSamp}_j(\mathsf{pk})$

Apply the steps listed in Algorithm 3 starting with Equation (21) and ending at Equation (22) to the state $|\phi\rangle = |0\rangle$ to obtain the state

$$X^{d\cdot(x_0\oplus x_1)+H(x_0)+H(x_1)} |0\rangle |d\rangle |y\rangle = \underbrace{|d \cdot (x_0 \oplus x_1) + H(x_0) + H(x_1)\rangle}_{k_j} |d\rangle |y\rangle .$$

Return (k_j, s_j) where $s_j := (d, y)$.

Algorithm 8 sSamp

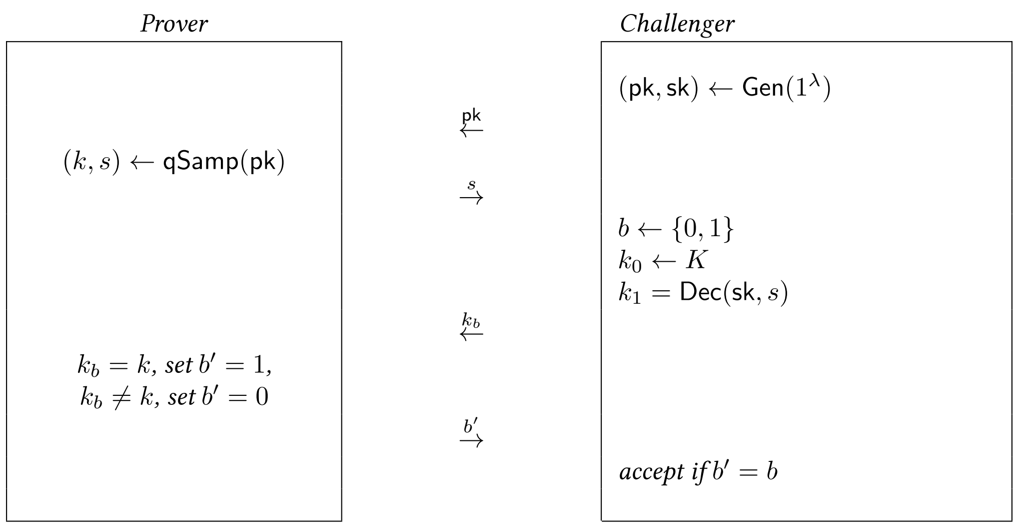
sSamp(pk)

- Sample $x \leftarrow \mathcal{X}$ (where $\mathcal{X}$ is specified by the NTCF $\mathcal{F}$ and the security parameter).
- Evaluate $y = \mathcal{F}.f_{\mathsf{pk},0}(x)$.
- Sample $d \leftarrow \mathcal{X} \setminus \{0\}$.
- Return (d, y).

In light of this, one can also consider the following primitive which implies an oblivious $\mathbf{U}$-pad using the ideas above.

Definition 23 (PoQ with efficient range sampling)**.** *A Proof of Quantumness with efficient range sampling is given by four algorithms* $(\mathsf{Gen}, \mathsf{qSamp}, \mathsf{Samp}, \mathsf{Dec})$ *where* qSamp *is QPT while* $\mathsf{Gen}, \mathsf{Samp}, \mathsf{Dec}$ *are PPT. Let* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ *and require that the following conditions hold.*

- *Correctness.* $k = \mathsf{Dec}(\mathsf{sk}, s)$ *for all* $(k, s) \leftarrow \mathsf{Enc}(\mathsf{pk})$.



- *Soundness. No PPT algorithm wins the security game above with probability more than* $1/2 + \mathsf{negl}(1/2)$.
- *Efficient range sampling condition. The distributions of* s *produced by* $(k, s) \leftarrow \mathsf{qSamp}(\mathsf{pk})$ *and* $s \leftarrow \mathsf{Samp}(\mathsf{pk})$ *are identical.*

9 Construction of the $(1, 1)$ compiler

With the Oblivious $\mathbf{U}$-Pad in place, we are now ready to describe our compiler. We start with the compiler for contextuality games where each context has size exactly 2, i.e. $|C| = 2$ for all $C \in C^{\text{all}}$. This subsumes 2-player non-local games as a special case. We describe the general compilers in Part III, but most of the main ideas already appear in the $|C| = 2$ case. In this section, we formally describe our compiler, and state its guarantees. In Section 10, we provide proofs.

Our compiler takes as input the following:

- A contextuality game $\mathsf{G} =: (Q, A, C^{\text{all}}, \text{pred}, \mathcal{D})$ with contexts of size 2, i.e. satisfying $|C| = 2$ for all $C \in C^{\text{all}}$.
- A strategy specified in terms of $\mathsf{qstrat} =: (\mathcal{H}, |\psi\rangle, \mathbf{O})$ for G (one that achieves the quantum value of G; used to describe the honest prover)
- A QFHE scheme as in Definition 5 and a security parameter λ.
  - Let $\mathbf{U} = \{U_k\}_{k\in K}$ be the group (up to global phases) of unitaries acting on $\mathcal{H}$, as in Equation (5).
  - Recall we use $\hat{k}$ to denote a classical encryption of k under the secret key of QFHE as in Section 3.2.
- An Oblivious $\mathbf{U}$-Pad scheme, OPad, as in Definition 19.

The compiler produces the following compiled game G' between a verifier and prover (summarised in Algorithm 9). We describe G', along with the actions of an honest prover that achieves the completeness guarantee.

1. The verifier proceeds as follows:
   (a) Sample a secret key $\mathsf{sk} \leftarrow \mathsf{QFHE.Gen}(1^\lambda)$, and a context $C \leftarrow \mathcal{D}$ according to the distribution specified by the game $\mathcal{D}$, picks a question q' from this context at random $q' \leftarrow C$. Evaluate $c_{q'} \leftarrow \mathsf{Enc}_{\mathsf{sk}}(q')$.
   (b) Sample a public key/secret key pair $(\mathsf{OPad.pk}, \mathsf{OPad.sk}) \leftarrow \mathsf{OPad}(1^\lambda)$.

   Send $(c_{q'}, \mathsf{OPad.pk})$ to the prover.
2. The honest prover prepares, under the QFHE encryption, the state $|\psi\rangle$ and subsequently measure $O_{q'}$ to obtain an answer a'. It then applies an oblivious $\mathbf{U}$-pad and returns the classical responses.
   (a) Since the operations happen under the QFHE encryption, the prover ends up with a QFHE encryption of a' which we denote by
   $$c_{a'}.$$
   Further, it holds a QFHE encryption of the post-measurement state which (by the assumption on the form of the QFHE encryption) looks like
   $$(U_{k''} |\psi_{q',a'}\rangle, \hat{k}''). \tag{26}$$
   Here $|\psi_{q',a'}\rangle$ is the post-measurement state when $|\psi\rangle$ is measured using $O_{q'}$ and the outcome a' is obtained.
   (b) The honest prover applies an oblivious $\mathbf{U}$-pad (see Algorithm 2) to obtain
   $$(U_{k'}U_{k''} |\psi_{q',a'}\rangle, s') \leftarrow \mathsf{OPad.Enc}(\mathsf{OPad.pk}, U_{k''} |\psi_{q',a'}\rangle).$$
   (c) The prover returns $(c_{a'}, \hat{k}'', s')$.
3. The verifier proceeds as follows:
   (a) Compute $a' := \mathsf{QFHE.Dec}_{\mathsf{sk}}(c_{a'})$, k'' from $\hat{k}''$ (the latter is by the form of QFHE) and $k' = \mathsf{OPad.Dec}(\mathsf{OPad.sk}, s')$.
   (b) Find k such that[24] $U_k = U_{k''}U_{k'}$ (such a k always exists because $\mathbf{U}$ form a group).

[24] up to a global phase

Algorithm 9 Game G' produced by the $(1,1)$-compiler for any contextuality game G with contexts of size two.

Honest Prover ($\mathcal{A}$)		Challenger ($\mathcal{C}$)
		$\mathsf{sk} \leftarrow \mathsf{QFHE.Gen}(1^\lambda)$ $C \leftarrow \mathcal{D}$ $q' \leftarrow C$ $c_{q'} \leftarrow \mathsf{QFHE.Enc}_{\mathsf{sk}}(q')$ $(\mathsf{OPad.pk}, \mathsf{OPad.sk}) \leftarrow \mathsf{OPad}(1^\lambda)$
	$\xleftarrow{(c_{q'},\mathsf{OPad.pk})}$	
Under the QFHE encryption, measures $O_{q'}$ and obtains an encrypted answer $c_{a'}$ and post-measurement state $(U_{k''}\left\vert\psi_{q'a'}\right\rangle, \hat{k}'')$. Applies an oblivious $\mathbf{U}$-pad to this state to obtain $(U_{k'}U_{k''}\left\vert\psi_{q'a'}\right\rangle, s') \leftarrow$ $\mathsf{OPad.Enc}(\mathsf{OPad.pk}, U_{k''}\left\vert\psi_{q'a'}\right\rangle)$.		
	$\xrightarrow{(c_{a'},\hat{k}'',s')}$	
		Using the secret keys sk, $\mathsf{OPad.sk}$, finds the k such that $U_k = U_{k''}U_{k'}$, samples $q \leftarrow C$
	$\xleftarrow{(q,k)}$	
Measures $U_k O_q U_k^\dagger$ and obtains a		
	$\xrightarrow{a}$	
		Computes $a' = \mathsf{Dec}_{\mathsf{sk}}(c_{a'})$. If $q = q'$, accept if $a = a'$ If $q \neq q'$, accept if $\mathrm{pred}((a,a'),C) = 1$.

(c) Samples a new question $q \leftarrow C$.

Send (q,k) in the clear to the prover.

4. The honest prover measures observable O_q conjugated by U_k, i.e. $U_k O_q U_k^\dagger$ and returns the corresponding answer a.

5. There are two cases, either the questions are the same or they are different (since $|C| = 2$). The verifier proceeds as follows:

   (a) If $q = q'$, accept if $a = a'$;

   (b) If $q \neq q'$, accept if $\mathrm{pred}((a,a'),C) = 1$.

We say that a prover wins G' if the verifier accepts.

9.1 Compiler Guarantees

The compiler satisfies the following.

Theorem 24 (Guarantees of the $(1,1)$ compiled contextuality game G')**.** *Suppose* QFHE *and* OPad *are secure (as in Definitions 5 and 19), and compatible (as in Definition 22). Let* G *be a contextuality game with* $\mathrm{valNC} < 1$ *and* $|C| = 2$ *for all contexts* $C \in C^{\mathrm{all}}$*. Let* G'_λ *be the compiled game produced by Algorithm 9 on input* G *and a security parameter* λ*. Then, the following holds.*

- *(Completeness) There is a negligible function* negl*, such that, for all* $\lambda \in \mathbb{N}$*, the honest QPT prover from Algorithm 9 wins* G'_λ *with probability at least*
$$c(\lambda) := \frac{1}{2}(1 + \mathrm{valQu}) - \mathsf{negl}(\lambda).$$
- *(Soundness) For every PPT adversary* $\mathcal{A}$*, there is a negligible function* negl' *such that, for all* $\lambda \in \mathbb{N}$*, the probability that* $\mathcal{A}$ *wins* G'_λ *is at most*
$$s(\lambda) := \frac{1}{2}(1 + \mathrm{valNC}) + \mathsf{negl}'(\lambda).$$

Furthermore, G'_λ *is faithful to* G *(as in Definition 18) with parameters* $s(\lambda)$ *and* $c(\lambda)$*.*

Completeness is straightforward to verify. We assume soundness for now and defer the proof to Section 10 (below). We outline how faithfulness in Definition 18 is satisfied.

Proof sketch. Parse the messages as follows,

$$\begin{aligned}(t_1, t_2) &= (c_{q'}, \mathsf{OPad.pk}),\\ (t_3, t_4, t_5) &= (c_{a'}, \hat{k}'', s'),\\ (t_6, t_7) &= (q, k), \text{ and}\\ t_8 &= a.\end{aligned}$$

The four criteria for faithfulness in Definition 18 are satisfied as follows.

1. *Well-formedness.* To satisfy the well-formedness property, we define the following.
   - $\{\mathsf{map}_i\}_i$ are naturally specified: $(\mathsf{map}_2, \mathsf{map}_4, \mathsf{map}_5, \mathsf{map}_7)$ output $\perp$ at all inputs, $(\mathsf{map}_6, \mathsf{map}_8)$ are identity maps (output whatever they are given as input) while $\mathsf{map}_1 = \mathsf{QFHE.Dec_{sk}}$ and $\mathsf{map}_3 := \mathsf{QFHE.Enc_{pk}}$. Under these maps, evidently, the verifier only asks questions in a single context, during any given execution.
   - qProver on input a description of $\mathsf{qstrat}' := (\mathcal{H}, |\psi'\rangle, \mathbf{O}')$, behaves as the honest prover in Algorithm 9 except that it uses qstrat' instead of the optimal state and measurement in qstrat. It is straightforward to note that qProver satisfies the "marginal" requirement.
2. G-*soundness.* This is straightforward to verify from the soundness value s and the construction of the compiler.
3. *Decision Faithfulness.* This is also straightforward to verify from the construction of the compiler.
4. G-*completeness.*
   (a) *Quantum completeness.*
   The first requirement (about existence of qstrat achieving c) follows directly from the fact that the completeness value is c and it is achieved by the honest prover in Algorithm 9.
   The second requirement (about the distribution of transcripts) is straightforward to verify by constructing a simulator $S_{\mathsf{qstrat,aux}}$ that behaves exactly like $\mathsf{qProver}(\mathsf{qstrat})$: it answers the questions corresponding G using qstrat and map_{ii} and aux specifies that the responses (t_4, t_5) are distributed exactly as $(\hat{k}'', s')$ produced by $\mathsf{qProver}(\mathsf{qstrat})$.
   (b) *Classical completeness.*
   Any qstrat' where all observables commute, may be seen as a distribution over truth tables. We show that P is a PPT machine that produces an identical transcript as $\mathsf{qProver}(\mathsf{qstrat})$. P proceeds as follows:
   - Sample a τ truth table according to the distribution above.
   - Receive $(c_{q'}, \mathsf{OPad.pk})$.

- Use QFHE.cEval to obtain $c_{\tau(q')}$ (see algorithm in the proof of Claim 6).
  Sample a random $r_{\tilde{k}} \leftarrow \mathsf{QFHE.Enc}(\tilde{k})$ for a uniform $\tilde{k} \leftarrow K$ and $r_s \leftarrow \mathsf{OPad.Samp}(\mathsf{pk})$ (see Algorithm 5).
  Send $(c_{\tau(q')}, r_{\tilde{k}}, r_s)$.
- Receive (q, k)
- Sends $\tau(q)$.

It is straightforward to verify that the transcript $(t_1, \ldots, t_8)$ produced by P is indeed identically distributed to that produced by qProver(qstrat) by using the classical range sampling property of OPad (see Definition 19) and the cEval property of QFHE (see Claim 6).

□

10 Soundness Analysis

The security notion we considered in the definition of our QFHE scheme (see Definition 5) says that encryptions of any two distinct messages is indistinguishable from one another. The corresponding security game is often referred to as 2-IND. What if one considers ℓ-IND, i.e. ℓ-many different encryptions? We prove that 2-IND implies a version of ℓ-IND which we call $\mathcal{D}$-IND'. $\mathcal{D}$-IND' is more general since it allows the ℓ messages to be sampled from an arbitrary distribution $\mathcal{D}$ (instead of being limited to uniform). $\mathcal{D}$-IND' is also more restricted in that (1) the ℓ messages are fixed and known to the challenger in advance and (2) we only consider classical algorithms.[25]

But why are we suddenly talking about ℓ-IND and $\mathcal{D}$-IND'? It turns out that the proof technique used to show 2-IND implies $\mathcal{D}$-IND' can be applied, albeit it takes some care, to prove the soundness of the compiler.

Thus, as a warmup, Section 10.1 shows that 2-IND implies $\mathcal{D}$-IND'. Then Section 10.2 shows how to apply these ideas to show that the 2-IND of QFHE encryptions implies soundness of the compiled game G' (using Algorithm 9) as stated in Theorem 24 (assuming that the OPad is secure). We emphasise that we do not use the result in Section 10.1 directly—we only use the proof technique.

10.1 Warm up | 2-IND implies $\mathcal{D}$-IND'

This subsection first recalls the 2-IND game (which is essentially a restatement of Equation (4)). Then it defines the $\mathcal{D}$-IND' game (the prime emphasises that the number of messages is constant and known apriori[26]). It ends by showing that 2-IND implies $\mathcal{D}$-IND'.

Claim 25 (2-IND Game for QFHE). Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ correspond to a QFHE encryption scheme. The scheme satisfies Equation (4) if and only if the following holds:
In the game below, for every PPT distinguisher, there is a negligible function negl such that the challenger accepts with probability at most $\frac{1}{2} + \mathsf{negl}(\lambda)$.

Distinguisher		Challenger
	$\xrightarrow{(x_0, x_1)}$	
		$\mathsf{sk} \leftarrow \mathsf{Gen}(1^\lambda)$ $b \leftarrow \{0,1\}$ $c_b \leftarrow \mathsf{Enc}_{\mathsf{sk}}(x_b)$
	$\xleftarrow{c_b}$	
Objective: guess b		
	$\xrightarrow{b'}$	
		Accept if $b' = b$.

Definition 26 ($\mathcal{D}$-IND' Game (where $\mathcal{D}$ is a distribution over a fixed message set) for QFHE)**.** *Let*

[25] It appears that proving the general version in the quantum case may not be completely straightforward because of the inability to apply rewinding directly.

[26] Proving the implication for the general case may be non-trivial in the quantum setting.

- *λ be a security parameter, $\ell = O(1)$ be a fixed constant (relative to λ),*
- *$M = \{m_1 \dots m_\ell\}$ be a set of distinct messages where $m_1, \dots m_\ell \in \{0,1\}^{O(1)}$ are of constant size,*
- *$\mathcal{D}$ be a probability distribution over M and let $q_{\text{guess}} = \max_i q_i$ where q_i is the probability assigned to m_i by by $\mathcal{D}$.*

The $\mathcal{D}$-IND' Security Game is a two-party game, for $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as specified by QFHE*, is as follows:*

Distinguisher		Challenger
		$\mathsf{sk} \leftarrow \mathsf{Gen}(1^\lambda)$ $m \leftarrow \mathcal{D}$ $c_m \leftarrow \mathsf{Enc}_{\mathsf{sk}}(m)$
	$\xleftarrow{c_m}$	
Objective: guess m		
	$\xrightarrow{m'}$	
		Accept if $m' = m$.

Proposition 27. *A* QFHE *scheme that satisfies Equation (4), implies that every PPT distinguisher for the $\mathcal{D}$-IND' game above, wins with probability at most $q_{\text{guess}} + \mathsf{negl}(\lambda)$.*

Proof. We can assume, without loss of generality, that the messages are given by $m_i = i$. This is because they can be uniquely indexed, and our arguments go through unchanged (as one can check).

Goal: we want to show that if there is a PPT distinguisher[27] $\mathcal{A}_\ell$ that wins against C_ℓ in the $\mathcal{D}$-IND' game above with probability $q_{\text{guess}} + \eta'$ for some non-negligible function η', then one can construct a PPT distinguisher $\mathcal{A}_2$ that wins against C_2 in the 2-IND game (see Claim 25) with probability at least $\frac{1}{2} + \eta''$ for some other non-negligible function η''.

To this end, denote by p_{ij} the probability that $\mathcal{A}_\ell$ outputs j when given the encryption of i as input, i.e.

$$p_{ij} := \Pr\left[j \leftarrow \mathcal{A}_\ell(c_i) : \begin{array}{c} \mathsf{sk} \leftarrow \mathsf{Gen}(1^\lambda) \\ c_i \leftarrow \mathsf{Enc}_{\mathsf{sk}}(i) \end{array} \right].$$

Since the message space is of constant size, $\mathcal{A}_2$ can compute p_{ij} to inverse polynomial errors. For now, we assume $\mathcal{A}_2$ knows p_{ij} exactly and handle the precision issue at the end. To proceed, consider the following observations:

1. $\Pr[\text{accept} \leftarrow \langle \mathcal{A}_\ell, C_\ell \rangle] = \sum_{i \in \{1 \dots \ell\}} q_i p_{ii}$ where recall that q_i is the probability assigned to i by the distribution $\mathcal{D}$.
2. There exist $k^* \neq i^*$ such that $\sum_j |p_{i^*j} - p_{k^*j}| \geq \eta$ for some non-negligible function η.

The first observation follows directly from the definition of $\mathcal{D}$-IND' and p_{ij}. The second follows from the assumption that $\mathcal{A}_\ell$ wins with probability at least $q_{\text{guess}} + \eta'$ for some non-negligible function η'. To see this, proceed by contradiction: Suppose for all $i^* \neq k^*$, it is the case that $\sum_j |p_{i^*j} - p_{k^*j}| \leq \text{negl}$ for some negligible function, then one could write

$$\begin{aligned}
\Pr[\text{accept} \leftarrow \langle \mathcal{A}_\ell, C_\ell \rangle] &= \sum_{i \in \{1 \dots \ell\}} q_i p_{ii} \\
&= \sum_{i \in \{1 \dots \ell\}} q_i (p_{i^*i} + \text{negl}) && \text{by the assumption above} \\
&\leq q_{\text{guess}} \cancelto{1}{\sum_{i \in \{1 \dots \ell\}} p_{i^*i}} + \text{negl} && \text{for another negl function} \\
&\leq q_{\text{guess}} + \text{negl}.
\end{aligned}$$

[27] Note that $\mathcal{A}_\ell$ and C_ℓ can depend on $\mathcal{D}$; the use of the subscript ℓ is just notational convenience and not meant to convey all the dependencies.

But this cannot happen because we assumed that $\mathcal{A}_\ell$ wins with probability non-negligibly greater than q_{guess}. We therefore conclude that observation 2 must hold. Since the message space is constant, a PPT $\mathcal{A}_2$ can determine the following:

- The indices $i^* \neq k^*$
- The disjoint sets $J_0, J_1 \subseteq \{1 \dots \ell\}$ such that
  - for $j \in J_0 \subseteq \{1 \dots \ell\}$, $p_{i^*j} \geq p_{k^*j}$, and
  - for $j \in J_1 \subseteq \{1 \dots \ell\}$, $p_{i^*j} < p_{k^*j}$.

Once these two indices, and the sets J_0, J_1 are known, $\mathcal{A}_2$'s remaining actions are as follows:

$\mathcal{A}_3$ (adversary for $\mathcal{D}$-IND')		$\mathcal{A}_2$ (the reduction)		$\mathcal{C}_2$ (challenger for 2-IND)
		Let $k^* \neq i^* \in \{1 \dots \ell\}$ be as above.	$\xrightarrow{(k^*, i^*) =: (x_0, x_1)}$	
				$\mathsf{sk} \leftarrow \mathsf{Gen}(1^\lambda)$ $b \leftarrow \{0,1\}$ $c_b \leftarrow \mathsf{Enc}_{\mathsf{sk}}(x_b)$
	$\xleftarrow{c_b}$		$\xleftarrow{c_b}$	
	$\xrightarrow{j_{\text{guess}}}$			
		$b' = \begin{cases} 0 & \text{if } j_{\text{guess}} = J_0 \\ 1 & \text{if } j_{\text{guess}} = J_1 \end{cases}$	$\xrightarrow{b'}$	
				Accept if $b' = b$.

Now,

$$\begin{aligned}
\Pr[\text{accept} \leftarrow \langle \mathcal{A}_2, \mathcal{C}_2 \rangle] =& \frac{1}{2} \cdot \sum_{j \in J_0} \Pr[\mathcal{A}_\ell \text{ outputs } j | i^* \text{ was encrypted}] + \\
& \frac{1}{2} \cdot \sum_{j \in J_1} \Pr[\mathcal{A}_\ell \text{ outputs } j | k^* \text{ was encrypted}] \\
=& \frac{1}{2} + \frac{1}{2} \sum_{j \in J_0} (p_{i^*j} - p_{k^*j}) && \text{By def of } p_{ij} \text{ and normalisation} \\
=& \frac{1}{2} + \frac{1}{4} \sum_{j} |p_{i^*j} - p_{k^*j}| && \|a-b\|_1 = 2 \sum_{i: a_i > b_i} a_i - b_i \text{ for } a, b \text{ distrib.} \qquad (27) \\
=& \frac{1}{2} + \frac{\eta}{4} && \text{from Observation 2.}
\end{aligned}$$

Since the QFHE scheme satisfies Equation (4), this is a contradiction (via Claim 25) and thus the claim follows—up to the precision issue which we now address.

Handling the precision issue. Denote by $\mathcal{A}_2$ the algorithm above where p_{ij} are known exactly. Suppose that

$$\Pr[\text{accept} \leftarrow \langle \mathcal{A}_\ell, C_\ell \rangle] \geq q_{\text{guess}} + \epsilon \tag{28}$$

where ϵ is a non-negligible function. For concreteness, suppose[28] $\epsilon(\lambda) = 1/\lambda^c$ for some constant $c > 0$.

Proposition 28. *Given that Equation (28) holds, consider an algorithm $\hat{\mathcal{A}}_2$ that uses an estimate for $\hat{p}_{ij}$ up to precision $O(\epsilon^3)$ (i.e. $|\hat{p}_{ij} - p_{ij}| \leq O(\epsilon^3)$). Then,*

$$\Pr[\text{accept} \leftarrow \langle \hat{\mathcal{A}}_2, C_2 \rangle] \geq \Pr[\text{accept} \leftarrow \langle \mathcal{A}_2, C_2 \rangle] - O(\epsilon^3). \tag{29}$$

To prove Proposition 28, we first show (see Claim 10.1 below) that $\sum_j |p_{i^*j} - p_{k^*j}|$ is at least $\Omega(\epsilon^2)$ given that Equation (28) holds. Using this and Equation (27), it follows that $\Pr[\text{accept} \leftarrow \langle \mathcal{A}_2, C_2 \rangle]$ is at least one half plus $\Omega(\epsilon^2)$ which means $\hat{\mathcal{A}}_2$—the finite-precision variant of $\mathcal{A}_2$—succeeds with one half plus non-negligible probability in the 2-IND game of the QFHE scheme (see Claim 25).

Claim. If Equation (28) holds then there exist $i^* \neq k^*$ such that $\|p_{i^*} - p_{j^*}\|_1 := \sum_j |p_{i^*j} - p_{k^*j}| \geq \Omega(\epsilon^2)$.

Proof of Claim 10.1. We start with an elementary fact: Let $f, g : \Lambda \to \mathbb{R}$, where $\Lambda \subset \mathbb{N}$ is an infinite subset of $\mathbb{N}$. Let P be the proposition that $f \leq O(g)$ on the set Λ. Then $\neg P$ implies that $f \geq \Omega(g)$ on an infinite set $\Lambda' \subseteq \Lambda$.

Using this fact, deduce that the negation of $\sum_j |p_{i^*j} - p_{k^*j}| \geq \Omega(\epsilon^2)$ implies there is some infinite set $\Lambda \subseteq \mathbb{N}$ over which $\sum_j |p_{i^*j} - p_{k^*j}| \leq O(\epsilon^2)$. We prove the claim by contradiction. Consider to the contrary that for all $i^* \neq k^*$, $\sum_j |p_{i^*j} - p_{k^*j}| < O(\epsilon^2)$. Then, it holds that

$$\begin{aligned}
\Pr[\text{accept} \leftarrow \langle \mathcal{A}_\ell, C_\ell \rangle] &= \sum_{i \in \{1 \dots \ell\}} q_i p_{ii} \\
&\leq \sum_{i \in \{1 \dots \ell\}} q_i (p_{i^*i} + O(\epsilon^2)) \\
&\leq q_{\text{guess}} \cancelto{1}{\sum_{i \in \{1 \dots \ell\}} p_{i^*i}} + O(\epsilon^2) && \because \ell = O(1) \\
&= q_{\text{guess}} + O(\epsilon^2)
\end{aligned}$$

but this violates Equation (28). $\square$

We can now prove Proposition 28.

Proof of Proposition 28. Since $\hat{\mathcal{A}}_2$ estimates p_{ij} to precision $O(\epsilon^3)$, it follows that it will find $i^* \neq k^*$ and a j such that $|p_{i^*j} - p_{k^*j}| \geq \Omega(\epsilon^2)$; recall that by the definition, $\ell = O(1)$ so $\|p_{i^*} - p_{j^*}\|_1 \geq \Omega(\epsilon^2)$ implies there is some j for which $|p_{i^*j} - p_{k^*j}| \geq \Omega(\epsilon^2)$.

Further, the only js for which $\hat{\mathcal{A}}_2$ makes an error in deciding whether to have $j \in J_0$, or $j \in J_1$, are those for which $|p_{i^*j} - p_{k^*j}| \leq O(\epsilon^3)$. (Note that it cannot be that for all js, $|p_{i^*j} - p_{k^*j}| \leq O(\epsilon^3)$ because then $\|p_{i^*} - p_{k^*}\|_1$ cannot be $\geq \Omega(\epsilon^2)$.) Thus, the error in computing $\Pr[\text{accept} \leftarrow \langle \hat{\mathcal{A}}_2, C_\ell \rangle]$ using $\Pr[\text{accept} \leftarrow \langle \mathcal{A}_2, C_\ell \rangle] = \frac{1}{2} + \frac{1}{2} \sum_{j \in J_0} (p_{i^*j} - p_{k^*j})$ is at most $O(\epsilon^3)$ (again, using $\ell = O(1)$). This yields Equation (29) as asserted. $\square$

This completes the proof. $\square$

10.2 The Reduction

We are going to essentially adapt the proof of Proposition 27 to our setting. Start by considering the honest prover $\mathcal{A}$ in Algorithm 9. We reduce $\mathcal{A}$ to a distinguisher $\mathcal{A}_2$ for the 2-IND security game of QFHE.

- Notation: Let (Q, A) denote the set of questions and answers in the contextuality game that was compiled.
- Phase 1: Learning "p_{ij}".
  - Consider the following procedure $\mathcal{A}_2.\mathsf{TruthTable}$ that takes $q' \in Q$ as input and produces a truth table $\tau : Q \to A$.

[28]For every non-negligible function ϵ, there is an infinite subset Λ of its domain where $\epsilon(\lambda) \geq 1/\lambda^c$. One can restrict the entire argument to this domain and still reach the same conclusion.

Algorithm 10 The procedure $\mathcal{A}_2$.TruthTable takes as input a question q' and produces a truth table τ corresponding to it. Note that this is a randomised procedure (depends on the QFHE encryption procedure) so for the same q' the procedure may output different τs. The goal is to learn the probabilities of different τs appearing for each question q'.

$\mathcal{A}$		$\mathcal{A}_2.\mathsf{TruthTable}(q')$
		$\mathsf{sk} \leftarrow \mathsf{QFHE.Gen}(1^\lambda)$ $c_{q'} \leftarrow \mathsf{QFHE.Enc}(q')$ $(\mathsf{OPad.pk}, \mathsf{OPad.sk}) \leftarrow \mathsf{OPad}(1^\lambda)$
	$\xleftarrow{(c_{q'},\mathsf{OPad.pk})}$	
	$\xrightarrow{(c_{a'},\hat{k}'',s')}$	
	$\circlearrowright$	Samples $k \leftarrow K$ For each $q \in Q$, ask (q,k), receive a and rewind until τ is fully specified.
		Outputs τ

1. $\mathcal{A}_2$ simulates the challenger $\mathcal{C}$ in Algorithm 9 as needed (see Algorithm 10), except that it takes q' as input (instead of uniformly sampling it in the beginning) and uses $k \leftarrow K$ (in the third step). It uses this simulation to generate the first message $(c_{q'}, \mathsf{OPad.pk})$ and feeds it to $\mathcal{A}$.
2. $\mathcal{A}_2$ receives $(c_{a'}, \hat{k}'', s')$ from $\mathcal{A}$. For each $q \in Q$, $\mathcal{A}_2$ sends (q, k) to $\mathcal{A}$ (where recall k is sampled uniformly at random from K) and receives an answer a. Denote by τ the truth table, i.e. the list of answers indexed by the questions.

- $\mathcal{A}_2$ repeats the procedure $\mathcal{A}_2 \cdot \mathsf{TruthTable}(q')$ above for each $q' \in Q$ to estimate the probability $p_{q'\tau}$ of the procedure outputting τ on input q' (the randomness is also over the encryption procedure etc). Here $p_{q'\tau}$ is analogous to p_{ij}.
- It finds questions $q_{*0} \neq q_{*1}$ such that

$$\|p_{q_{*0}} - p_{q_{*1}}\| := \sum_\tau |p_{q_{*0}\tau} - p_{q_{*1}\tau}| \geq \eta \tag{30}$$

  for some non-negligible function η.
  (We defer the proof that questions (or indices) $q_{*0} \neq q_{*1}$ exist if the OPad is secure and $\mathcal{A}$ wins the compiled contextuality game G', with probability non-negligibly greater than $\frac{1}{2}(1 + \mathrm{valNC})$.)
- It defines the disjoint sets T_0 and T_1 as follows: $\tau \in T_0$ if $p_{q_{*0}\tau} \geq p_{q_{*1}\tau}$ and $\tau \in T_1$ if $p_{q_{*1}\tau} > p_{q_{*0}\tau}$.

• Phase 2: Interaction with C_2 of the 2-IND game.
  - $\mathcal{A}_2$ sends q_{*0}, q_{*1} to C_2 and C_2 returns c_b, the QFHE encryption of q_{*b} (where C_2 picks $b \leftarrow \{0, 1\}$).
  - $\mathcal{A}_2$ simulates the OPad itself and forwards c_b to $\mathcal{A}$, learns the truth table τ corresponding to c_b and outputs $b' \in \{0, 1\}$ such that $\tau \in T_{b'}$.

  The intuition is that given an encryption of q_{*0}, on an average, the above procedure would output $b' = 0$ more often than $b' = 1$, essentially by definition of T_0 and T_1, and Equation (30).

Remark 29. There are three subtleties that are introduced, aside from the rewinding needed to learn τ, in analysing G' as opposed to the $\mathcal{D}$-IND' game. We glossed over these above.

1. *Random* k: When interacting with the prover $\mathcal{A}$, $\mathcal{A}_2$ above is feeding in a uniformly random k instead of the correct k which depends on $\hat{k}''$ and s' (see Algorithm 9 where $\mathcal{A}$ interacts with $\mathcal{C}$ and compare it to the interaction of $\mathcal{A}$ with $\mathcal{A}_2$).

Algorithm 11 The algorithm $\mathcal{A}_2$ uses the adversary $\mathcal{A}$ for the compiled contextuality game G' to break the 2-IND security game for the QFHE scheme.

$\mathcal{A}$ $\longleftrightarrow$ $\mathcal{A}_2$

Phase 1
Compute $p_{q'\tau}$ (as in the description)
for each $q' \in Q$ by
running $\mathcal{A}_2.\mathsf{TruthTable}(q')$
(see point 3, Remark 29).

Use $p_{q'\tau}$ to learn
the questions q_{*0}, q_{*1} and
the disjoint sets T_0, T_1 of truth
tables.

$\mathcal{A}$ | $\mathcal{A}_2$ | C_2

Phase 2

$\mathcal{A}_2$: $(\mathsf{OPad.pk}, \mathsf{OPad.sk}) \leftarrow \mathsf{OPad}(1^\lambda)$

$\mathcal{A}_2 \xrightarrow{(q_{*0}, q_{*1})} C_2$

C_2: $\mathsf{sk} \leftarrow \mathsf{QFHE.Gen}(1^\lambda)$
$b \leftarrow \{0,1\}$
$c_b \leftarrow \mathsf{QFHE.Enc}_{\mathsf{sk}}(q_{*b})$

$\mathcal{A}_2 \xleftarrow{c_b} C_2$

$\mathcal{A} \xleftarrow{(c_b, \mathsf{OPad.pk})} \mathcal{A}_2$

$\mathcal{A} \xrightarrow{(c_{a'}, \hat{k}'', s')} \mathcal{A}_2$

$\circlearrowleft$

$\mathcal{A}_2$: Samples $k \leftarrow K$
For each $q \in Q$, repeats (q, k) to
determine τ

Set $b' = 0$ if $\tau \in T_0$,
and $b' = 1$ if $\tau \in T_1$

$\mathcal{A}_2 \xrightarrow{b'} C_2$

C_2: Accept if $b' = b$

Algorithm 12 Whether a PPT adversary $\mathcal{A}$ for the compiled contextuality game G' is used with the correct k or a uniformly random k does not affect the outcome of this interaction more than negligibly.

$\mathcal{A}$		$B_0(q')$ (resp. $B_1(q')$)
		$\mathsf{sk} \leftarrow \mathsf{QFHE.Gen}(1^\lambda)$ $c_{q'} \leftarrow \mathsf{QFHE.Enc}_{\mathsf{sk}}(q')$ $(\mathsf{OPad.pk}, \mathsf{OPad.sk}) \leftarrow \mathsf{OPad}(1^\lambda)$
	$\xleftarrow{(c_{q'}, \mathsf{OPad.pk})}$	
	$\xrightarrow{(c_{a'}, \hat{k}'', s')}$	
		B_0 computes $k' := \mathsf{OPad.Dec}(\mathsf{OPad.sk}, s')$ and uses the secret key sk to compute k'' and then finds the k satisfying $U_k = U_{k''}U_{k'}$. (resp. B_1 samples a uniform $k \leftarrow K$).
	(q,k) $\circlearrowleft$	Potentially rewinds $\mathcal{A}$ to this step and queries with (q, k) for arbitrary $q \in Q$. Runs an arbitrary procedure to compute a bit b'.

- However, the guarantees about $\mathcal{A}$ are for the correct k.
- We need to formally show that the security of OPad allows us to work with random ks directly.
- This is straightforward enough but we need to use this a few times; we'll see.

2. *$\mathcal{A}$ is consistent*: The other subtlety has to do with the proof that $\mathcal{A}$ winning with probability non-negligibly more than $\frac{1}{2}(1 + \mathrm{valNC})$ implies $\|p_{q_{*0}} - p_{q_{*1}}\|_1$ is non-negligible. In the proof, we use the assumption that $\mathcal{A}$ is *consistent*, i.e. if it answers with an encryption of a' upon being asked an encryption of q' in the first interaction, it will respond consistently if the same question is asked in the clear, i.e. it answers a' upon being asked the same question q' in the next round. We will show that given an $\mathcal{A}$ that is non-consistent, one can still bound it's success probability by treating it as though it is consistent.

3. *Precision of $p_{q'\tau}$*: We also completely skipped the precision issue, i.e. we assumed $\mathcal{A}_2$ can learn $p_{q'\tau}$ exactly. In practice, this is not possible, of course. However, $p_{q'\tau}$ can be learnt to enough precision to make the procedure work, just as we did in the proof of Proposition 27.

10.3 Proof Strategy

Let $\mathcal{A}$ be any PPT algorithm interacting with $\mathcal{C}$ in the compiled contextuality game G'.

The following allows us to assume that we can give a uniformly random k as input to $\mathcal{A}$ without changing the output distribution of the interaction in Algorithm 12 (as alluded to in point 1 of Remark 29).

Lemma 30 (Uniformly random k is equivalent to the correct k)**.** *Let B_0 (resp. B_1) be a PPT algorithm that takes $q' \in Q$ as an input, interacts with $\mathcal{A}$ and outputs a bit, as described in Algorithm 12. Then there is a negligible function* negl *such that* $|\Pr[0 \leftarrow \langle B_0, \mathcal{A}\rangle] - \Pr[0 \leftarrow \langle B_1, \mathcal{A}\rangle]| \leq \mathsf{negl}$.

The following allows us to treat $\mathcal{A}$ as though it is consistent (as anticipated in point 2 of Remark 29).

Lemma 31 (Consistency only helps)**.** *Let*

- $\mathcal{C}_{k\leftarrow K}$ *be exactly the same as the challenger* $\mathcal{C}$ *for the compiled contextuality game* G' *except that it samples a uniformly random* k *instead of computing it correctly, let*
- $\mathcal{A}$ *be any PPT algorithm that is designed to play the complied contextuality game* G' *and makes* $\mathcal{C}_{k\leftarrow K}$ *accept with probability* p*, i.e.* $\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C}_{k\leftarrow K}\rangle] = p$*, and denote by*
- $p_{q'\tau}$ *the probability that on being asked* q'*, the truth table* $\mathcal{A}$ *produces is* τ *(as defined in Section 10.2).*

Then

$$p \leq \sum_C \Pr(C) \cdot \frac{1}{2}\left(1 + \frac{1}{|C|}\sum_{q'\in C}\sum_{\tau} p_{q'\tau}\mathsf{pred}(\tau[C], C)\right)$$

where $\Pr(C)$ *denotes the probability with which* $\mathcal{C}$ *samples the context* C*.*

The two terms in the sum above, correspond to the consistency test ($q = q'$) and the predicate test ($q' \neq q$). Finally, the following allows us to neglect the precision issue (as detailed in point 3 of Remark 29).

Lemma 32 (Precision is not an issue)**.** *Suppose*

- $\mathcal{A}$ *wins with probability* $\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C}\rangle] \geq \frac{1}{2}(1 + \mathrm{valNC}) + \epsilon$ *for some non-negligible function* ϵ
- $\mathcal{A}_2$ *is as in Algorithm 11, i.e. it is a PPT algorithm except for the time it spends in learning* $p_{q'\tau}$ *exactly*
- *Denote by* $\mathcal{A}_{2,\epsilon}$ *a PPT algorithm that is the same as* $\mathcal{A}_2$ *except that it computes and uses an estimate* $\hat{p}_{q'\tau}$ *satisfying* $|\hat{p}_{q'\tau} - p_{q'\tau}| \leq O(\epsilon^3)$*, in place of* $p_{q'\tau}$*.*

Then, the PPT algorithm $\mathcal{A}_{2,\epsilon}$ *wins with essentially the same probability as* $\mathcal{A}_2$*, i.e.*

$$\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}_{2,\epsilon}, C_2\rangle] \geq \Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}_2, C_2\rangle] - O(\epsilon^3).$$

We prove the following contrapositive version of the soundness guarantee in Theorem 24.

Theorem 33 (Soundness condition restated from Theorem 24)**.** *Suppose*

- $\mathcal{A}$ *is any PPT algorithm that wins with probability* $\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C}\rangle] \geq \frac{1}{2}(1 + \mathrm{valNC}) + \epsilon$ *for some non-negligible function* ϵ*, and*
- *the* OPad *used is secure, then*

there is a PPT algorithm $\mathcal{A}_{2,\epsilon}$ *that wins the* 2*-IND security game of the* QFHE *scheme with probability*

$$\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}_{2,\epsilon}, C_2\rangle] \geq \frac{1}{2} + \mathsf{nonnegl},$$

where $\mathsf{nonnegl}$ *is a non-negligible function that depends on* ϵ*.*

In Section 10.4, we prove Theorem 33 assuming Lemmas 30, 31 and 32. In Section 10.5, we prove the lemmas.

10.4 Proof assuming the lemmas (Step 1 of 2)

Proof. This part is analogous to the proof of Proposition 27. First, recall the challenger $\mathcal{C}$ of the complied game G' and let $\mathcal{C}_{k\leftarrow K}$ denote the the same challenger, except that it samples k uniformly at random. From Lemma 30, one can conclude that $|\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C}\rangle] - \Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C}_{k\leftarrow K}\rangle]| \leq \mathsf{negl}$.

Recall the definition of $p_{q'\tau}$ from the discussion in Section 10.2. Using Lemma 31, one can write

$$\begin{aligned}\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C}\rangle] - \mathsf{negl} &\leq \sum_C \Pr(C) \cdot \frac{1}{2}\left(1 + \frac{1}{|C|}\sum_{q'\in C}\sum_{\tau} p_{q'\tau}\mathsf{pred}(\tau[C], C)\right) \\ &= \frac{1}{2}\left(1 + \sum_C \Pr(C)\frac{1}{|C|}\sum_{q'\in C}\sum_{\tau} p_{q'\tau}\mathsf{pred}(\tau[C], C)\right) \qquad (31)\end{aligned}$$

Observe also that there exist $q_{*0} \neq q_{*1}$ such that

$$\|p_{q_{*0}} - p_{q_{*1}}\|_1 := \sum_{\tau} |p_{q_{*0}\tau} - p_{q_{*1}\tau}| \geq \eta \tag{32}$$

for some non-negligible function η. This is a consequence of the assumption that

$$\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C} \rangle] \geq \frac{1}{2}(1 + \mathrm{valNC}) + \epsilon \tag{33}$$

for some non-negligible function ϵ. To see this, proceed by contradiction: Suppose that for all $q_{*0} \neq q_{*1}$, it is the case that $\sum_{\tau} |p_{q_{*0}\tau} - p_{q_{*1}\tau}| \leq \mathsf{negl}$ for some negligible function, then one could write, using Equation (31),

$$\begin{aligned}
\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C} \rangle] &\leq \frac{1}{2}\left(1 + \sum_{C} \Pr(C) \cancel{\frac{1}{|C|} \sum_{q' \in C}} \sum_{\tau} p_{q_{*0}\tau} \mathsf{pred}(\tau[C], C)\right) + \mathsf{negl}' \\
&= \frac{1}{2}\left(1 + \sum_{\tau} p_{q_{*0}\tau} \sum_{C} \Pr(C) \mathsf{pred}(\tau[C], C)\right) + \mathsf{negl}' \\
&\leq \frac{1}{2}(1 + \mathrm{valNC}) + \mathsf{negl}'
\end{aligned} \tag{34}$$

where in the first step, we used $p_{q_{*0}\tau}$ instead of $p_{q'\tau}$ at the cost of a negl' additive error, in the second step, we re-arranged the sum, and in the final step, we observe that the expression is just a convex combination of values achieved using non-contextual strategies, and this is at most valNC. However, Equation (34) contradicts Equation (33) that says, by assumption, $\mathcal{A}$ wins with probability non-negligibly more than $\frac{1}{2}(1 + \mathrm{valNC})$.

So far, we have established Equation (32) holds for some distinct questions $q_{*0} \neq q_{*1}$. We assume that $\mathcal{A}_2$ can learn $p_{q'\tau}$ exactly and invoke Lemma 32 to handle the fact that these can only be approximated to inverse polynomial errors but that does not change the conclusion. Therefore, $\mathcal{A}_2$ can learn $q_{*0} \neq q_{*1}$ and construct the sets T_0, T_1 of truth tables (recall: τ is in T_0 if $p_{q_{*0}\tau} \geq p_{q_{*1}\tau}$ and in T_1 otherwise). Proceeding as in the proof of Proposition 27, and focusing on Phase 2 of the interaction, it holds that

$$\begin{aligned}
\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}_2, C_2 \rangle] = &\frac{1}{2} \cdot \sum_{\tau \in T_0} \Pr[\mathcal{A} \text{ outputs } \tau | q_{*0} \text{ was encrypted}] + \\
&\frac{1}{2} \cdot \sum_{\tau \in T_1} \Pr[\mathcal{A} \text{ outputs } \tau | q_{*1} \text{ was encrypted}] \\
&= \frac{1}{2} + \frac{1}{2} \sum_{\tau \in T_0} (p_{q_{*0}\tau} - p_{q_{*1}\tau}) \\
&= \frac{1}{2} + \frac{1}{4} \|p_{q_{*0}} - p_{q_{*1}}\|_1 \\
&= \frac{1}{2} + \frac{\eta}{4}.
\end{aligned}$$

Since the QFHE scheme satisfies Equation (4), we have a contradiction (via Claim 25) which means our assumption that $\mathcal{A}$ wins with probability non-negligibly more than $\frac{1}{2}(1 + \mathrm{valNC})$ is false, completing the proof. ☐

10.5 Proof of the lemmas (Step 2 of 2)

Lemma 30, about using a uniformly random k instead of the correct one, is almost immediate but we include a brief proof.

Proof of Lemma 30. Consider the adversary $\mathcal{A}_{\mathsf{OPad}}$ for OPad as defined in Algorithm 13.

Algorithm 13 Whether a PPT adversary $\mathcal{A}$ for the compiled contextuality game G' is used with the correct k or a uniformly random k, it makes no difference, if all algorithms involved are PPT.

$\mathcal{A}$		$\mathcal{A}_{\mathsf{OPad}}$		C_{OPad}
		$\mathsf{sk} \leftarrow \mathsf{QFHE.Gen}(1^\lambda)$ $c_{q'} \leftarrow \mathsf{QFHE.Enc}_{\mathsf{sk}}(q')$		$(\mathsf{OPad.pk}, \mathsf{OPad.sk}) \leftarrow \mathsf{OPad}(1^\lambda)$
	$\xleftarrow{(c_{q'}, \mathsf{OPad.pk})}$		$\xleftarrow{\mathsf{OPad.sk}}$	
	$\xrightarrow{(c_{a'}, \hat{k}'', s')}$		$\xrightarrow{s'}$	
				$b \leftarrow \{0,1\}$, $k_0 \leftarrow K$, $k_1 = \mathsf{OPad.Dec}(\mathsf{OPad.sk}, s')$
			$\xleftarrow{k_b}$	
	(q,k) $\circlearrowleft$ a	B defines $k' := k_b$ and uses the secret key sk to compute k'' and then finds the k satisfying $U_k = U_{k''} U_{k'}$. Potentially rewinds $\mathcal{A}$ to this step and queries with (q,k) for arbitrary $q \in Q$. Runs an arbitrary procedure to compute a bit b'.		
			$\xrightarrow{b'}$	
				Accept if $b' = b$

Observe that

$$\begin{aligned}
\Pr\left[\mathsf{accept} \leftarrow \langle \mathcal{A}_{\mathsf{OPad}}, C_{\mathsf{OPad}} \rangle\right] =& \frac{1}{2} \Pr[\mathcal{A}_{\mathsf{OPad}} \text{ outputs } b' = 0 | b = 0] + \\
& \frac{1}{2} \Pr[\mathcal{A}_{\mathsf{OPad}} \text{ outputs } b' = 1 | b = 1] \\
=& \frac{1}{2} \Big(\Pr[\mathcal{A}_{\mathsf{OPad}} \text{ outputs } b' = 0 | b = 0] - \\
& \Pr[\mathcal{A}_{\mathsf{OPad}} \text{ outputs } b' = 0 | b = 1] \Big) + \frac{1}{2} \\
=& \frac{1}{2} \left(\Pr[0 \leftarrow \langle B_0, \mathcal{A} \rangle] - \Pr[0 \leftarrow \langle B_1, \mathcal{A} \rangle] \right) + \frac{1}{2}
\end{aligned} \tag{35}$$

where B_0 and B_1 interact with $\mathcal{A}$ as in Algorithm 12. The last equality holds because for a random k', k also becomes random.

Let $\mathcal{A}'_{\mathsf{OPad}}$ be $\mathcal{A}_{\mathsf{OPad}}$ except that it outputs $b' \oplus 1$ instead of b'. Using the analogue of Equation (35) for $\mathcal{A}'_{\mathsf{OPad}}$, Equation (35) itself and the security of OPad, it follows that there is a negligible function negl such that

$$|\Pr[0 \leftarrow \langle B_0, \mathcal{A} \rangle] - \Pr[0 \leftarrow \langle B_1, \mathcal{A} \rangle]| \leq \mathsf{negl}.$$

□

We now look at the proof of Lemma 31 which crucially relies on the fact that the consistency test and the predicate test happen with equal probability.

Proof of Lemma 31. Consider the adversary $\mathcal{A}'$ in Algorithm 14 that uses $\mathcal{A}$ to interact with $\mathcal{C}_{k \leftarrow K} =: \mathcal{C}'$.

Algorithm 14 $\mathcal{A}'$, a potentially QPT algorithm, uses the PPT algorithm $\mathcal{A}$ to play the compiled contextuality game G'. Its winning probability upper bounds that of $\mathcal{A}$ and can be computed in terms of $p_{q'\tau}$ for $\mathcal{A}$.

$\mathcal{A}$ | $\mathcal{A}'$ | $\mathcal{C}_{k\leftarrow K} =: \mathcal{C}'$

$\mathcal{A} \xleftarrow{(c_{q'},\mathsf{OPad.pk})} \mathcal{A}' \xleftarrow{(c_{q'},\mathsf{OPad.pk})} \mathcal{C}'$

$\mathcal{A} \xrightarrow{(c_{a'},\hat{k}'',s')} \mathcal{A}'$

$\mathcal{A}$: $(\tilde{q},\tilde{k})$ ⟳

$\mathcal{A}'$: $\tilde{k} \leftarrow K$

Rewinds $\mathcal{A}'$ to learn the truth table τ.

Evaluate $c_{\tau(q')} \leftarrow \mathsf{QFHE.Eval}_\tau(c_{q'})$.

$\mathcal{A}' \xrightarrow{(c_{\tau(q')},\hat{k}'',s')} \mathcal{C}'$

$\mathcal{A}' \xleftarrow{(q,k)} \mathcal{C}'$

$\mathcal{A}' \xrightarrow{a:=\tau(q)} \mathcal{C}'$

$\mathcal{C}'$: Recall:
$(a' = \mathsf{Dec}_{\mathsf{sk}}(c_{\tau(q')}))$
If $q = q'$, accept if $a = a'$.
If $q \neq q'$, accept if $\mathrm{pred}((a,a'),C) = 1$.

We show that

$$\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C}' \rangle] \le \Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}', \mathcal{C}' \rangle] \tag{36}$$

$$= \sum_C \Pr(C) \cdot \frac{1}{2} \left(1 + \frac{1}{|C|} \sum_{q' \in C} \sum_\tau p_{q'\tau} \mathsf{pred}(\tau[C], C) \right). \tag{37}$$

We note that $\mathcal{A}'$ may be a QPT algorithm because it runs $\mathsf{QFHE.Eval}$. Indeed, existing QFHE schemes don't produce a classical procedure for $\mathsf{QFHE.Eval}$, even when the ciphertext and the logical circuit are classical. However, we only use $\mathcal{A}'$ to compute an upper bound on the performance of $\mathcal{A}$ in terms of $p_{q'\tau}$ and therefore $\mathcal{A}'$ being QPT (as opposed to PPT) is not a concern here.

We first derive Equation (36). Consider the interactions $\langle \mathcal{A}, \mathcal{C}' \rangle$ and $\langle \mathcal{A}', \mathcal{C}' \rangle$. Note that for any given $c_{q'}$, the challenger $\mathcal{C}'$ either asks $q = q'$ or $q \neq q'$ with equal probabilities. Conditioned on $c_{q'}$, there are two cases: (1) $\mathcal{A}$ is consistent, in which case the answers to q, q' given by $\mathcal{A}'$ and $\mathcal{A}$ are identical. (2) $\mathcal{A}$ is inconsistent, in which case, $\mathcal{C}'$ accepts $\mathcal{A}$ with probability *at most* $1/2$ while $\mathcal{C}'$ accepts $\mathcal{A}'$ with probability *at least* $1/2$.

Equation (37) follows because $\mathcal{C}'$ selects a context C with probability $\Pr(C)$, the $1/2$ denotes whether a consistency test ($q = q'$) is performed or a predicate test ($q \neq q'$) is performed. By construction of $\mathcal{A}'$, the consistency test passes with probability 1. The $1/|C|$ factor indicates that either of the two questions in C could have been asked as the first question q' (under the QFHE encryption; and $|C| = 2$ here). Again, by construction of $\mathcal{A}'$, the probability of clearing the predicate test is the weighted average of $\mathsf{pred}(\tau[C], C)$ where the weights are given by $p_{q'\tau}$. This completes the proof. □

Lemma 32 follows by proceeding as in the proof of Proposition 28.

Part III

General Computational Test of Contextuality—beyond size-2 contexts

So far, we looked at contextuality games with size 2 contexts. This part explains how to generalise our compiler to games with contexts of arbitrary size. In fact, we consider two generalisations of our compiler, both of which produce single prover, 4-message (2-round) compiled games, irrespective of the size of contexts in the original contextuality game.

The $(|C|, 1)$ compiler. The compiled game asks all $|C|$ questions from one context under the QFHE encryption in the first round, and asks one question in the clear in the second round. It guarantees that no PPT algorithm can succeed with probability more than

$$1 - \mathsf{const}_1 + \mathsf{negl}$$

where $\mathsf{const}_1 = 1/|Q|$ when all questions are asked uniformly at random.[29] The proof idea is similar to the $(1, 1)$ compiler with one major difference—the analogue of the bound in Lemma 31 changes. In essence, there one could obtain the bound by constructing a prover that is always consistent (i.e. the encrypted answer and the answer in the clear match when the corresponding questions match). Here, we have the "dual" property instead. The bound is obtained by constructing a prover that always satisfies the constraint but may not be consistent.
As stated in the introduction, while this compiler works for games with perfect completeness, the bound on the classical value is sometimes more than the honest quantum value—so this compiler fails to give a separation between quantum and classical for some games (including the KCBS game of Example 3). However, for other games (like the magic square game of Example 1, it yields a larger completeness-soundness gap than the following universal compiler.

The $(|C| - 1, 1)$ compiler. This compiler addresses the limitation of the one above and is truly universal—in the sense that for any contextuality game G with $\mathrm{valNC} < \mathrm{valQu}$, the compiled game G' will also be such that every PPT algorithm wins with probability strictly smaller than a QPT algorithm. More precisely, PPT algorithms cannot succeed with probability more than

$$\frac{1}{|C|}\left(|C| - 1 + \mathrm{valNC}\right) + \mathsf{negl}$$

while there is a QPT algorithm that wins with probability at least

$$\frac{1}{|C|}\left(|C| - 1 + \mathrm{valQu}\right) - \mathsf{negl}.$$

Note that for $|C| = 2$, we recover

$$\frac{1}{2}\left(1 + \mathrm{valNC}\right) + \mathsf{negl}, \quad \frac{1}{2}(1 + \mathrm{valQu}) - \mathsf{negl}$$

respectively, which are the bounds for our $(1, 1)$ compiler.
The idea behind the construction is the following:

- Sample a context C and pick a question $q_{\mathsf{skip}} \leftarrow C$ uniformly at random.
- Ask all questions in C except q_{skip}, i.e. ask $C \backslash q_{\mathsf{skip}}$, under QFHE encryption
- Sample $q \leftarrow C$ and ask q in the clear.
- Now,
  1. if $q = q_{\mathsf{skip}}$, test the predicate using the decrypted answers to $C \backslash q_{\mathsf{skip}}$ and to q;

[29] $\mathsf{const}_1 = \min_{C \in C^{\mathrm{all}}} \Pr(C)/|C|$ in general.

2. if $q \neq q_{\mathsf{skip}}$, and so $q \in C \backslash q_{\mathsf{skip}}$; check that the corresponding answers are consistent.

The proof is based on the following idea.

- The key observation is that, just as in the $(1, 1)$ compiler, given an adversary $\mathcal{A}$, one can consider an adversary $\mathcal{A}'$ such that it is "consistent" and wins with at least as much probability as $\mathcal{A}$ and using this, one can obtain a bound on the success probability of $\mathcal{A}$ in terms of valNC.
- The compiled game was purposefully designed to ensure the following: being "consistent" can only help. As we saw, this was not the case for the $(|C|, 1)$ game—so selecting fewer questions in the first round is what, perhaps surprisingly, allows us to construct a universal test (in the sense described above).

Before moving to formal descriptions and proofs, we remark that it would be nice to have the success probability be independent of $|C|$, but this somehow seems hard to avoid. Why? Because the dependence on $|C|$ comes essentially from the fact the we are only asking one question in the clear in the second round. So, it seems that to avoid this dependence one would have to ask more than one question in the clear. However, this complicates the task of extracting a non-contextual assignment by rewinding the PPT adversary, because, for instance, the prover can now assign different values to q_1 depending on whether it was asked with q_2 or q_3. Thus, obtaining a better compiler seems to require new ideas.

11 Construction of the $(|C|, 1)$ compiler

We define the compiler formally first.

Algorithm 15 Game G' produced by the $(|C|, 1)$-compiler on input a contextuality game G, and security parameter λ.

Honest Prover ($\mathcal{A}$)		Challenger ($\mathcal{C}$)
		$\mathsf{sk} \leftarrow \mathsf{QFHE.Gen}(1^\lambda)$ $C \leftarrow \mathcal{D}$ $c \leftarrow \mathsf{QFHE.Enc}_{\mathsf{sk}}(C)$ $(\mathsf{OPad.pk}, \mathsf{OPad.sk}) \leftarrow \mathsf{OPad}(1^\lambda)$
	$\xleftarrow{(c,\mathsf{OPad.pk})}$	
Under the QFHE encryption, measures $\{O_q\}_{q \in C}$ and obtains an encrypted answers $\{c_{\mathbf{a}}\}$ (where $\mathbf{a}$ are the answers indexed by C) and post-measurement state $(U_{k''} \lvert \psi_{\mathbf{a}C} \rangle, \hat{k}'')$. Applies an oblivious $\mathbf{U}$-pad to this state to obtain $(U_{k'} U_{k''} \lvert \psi_{\mathbf{a}C} \rangle, s') \leftarrow \mathsf{OPad.Enc}(\mathsf{OPad.pk}, U_{k''} \lvert \psi_{\mathbf{a}C} \rangle)$.		
	$\xrightarrow{(c_{\mathbf{a}},\hat{k}'',s')}$	
		Using the secret keys sk, & $\mathsf{OPad.sk}$, finds the k such that $U_k = U_{k''} U_{k'}$, samples $q \leftarrow C$
	$\xleftarrow{(q,k)}$	
Measures $U_k O_q U_k^\dagger$ and obtains a		
	$\xrightarrow{a}$	
		Computes $\mathbf{a} = \mathsf{Dec}_{\mathsf{sk}}(c_{\mathbf{a}})$. Accepts if both (1) and (2) hold: (1) $\mathsf{pred}(\mathbf{a}, C) = 1$, and (2) $\mathbf{a}[q] = a$.

11.1 Compiler Guarantees

The compiler satisfies the following.

Theorem 34 (Guarantees of the $(|C|, 1)$ compiled contextuality game G')**.** *Suppose* QFHE *and* OPad *are secure (as in Definitions 5 and 19), and compatible (as in Definition 22). Let* G *be any contextuality game with* $\mathrm{valNC} < 1$*. Let* G'_λ *be the compiled game produced by Algorithm 15 on input* G *and a security parameter* λ*. Then, the following holds.*

- *(Completeness) There is a negligible function* negl*, such that, for all* $\lambda \in \mathbb{N}$*, the honest QPT prover from Algorithm 15 wins* G'_λ *with probability at least*

$$c(\lambda) := \mathrm{valQu} - \mathsf{negl}(\lambda).$$

- *(Soundness) For every PPT adversary $\mathcal{A}$, there is a negligible function* negl' *such that, for all $\lambda \in \mathbb{N}$, the probability that $\mathcal{A}$ wins G'_λ is at most*
$$s(\lambda) := 1 - \mathsf{const}_1 + \mathsf{negl}'(\lambda)\,,$$
where $\mathsf{const}_1 = \min_{C \in C^{\mathrm{all}}} \Pr(C)/|C| = O(1)$.

Furthermore, G'_λ is faithful to G (as in Definition 18) with parameters $s(\lambda)$ and $c(\lambda)$.

Completeness is straightforward to verify. The proof of faithfulness is analogous to that of Theorem 24. We prove soundness in Section 12.

Note that in games where all questions are asked uniformly at random, $\mathsf{const}_1 = 1/(|C^{\mathrm{all}}| \cdot |C|)$, yielding the simple bound $1 - 1/(|C^{\mathrm{all}}| \cdot |Q|) + \mathsf{negl}$ for PPT provers (that we quoted earlier).

Two brief remarks about the theorem are in order before we give the proof. First, we note that unlike the completeness value, which is $\mathrm{valQu} - \mathsf{negl}$, the soundness does not correspondingly depend on valNC. Second, as mentioned earlier, this compiler is not universal: for KCBS the classical value in G' is 0.9 which is greater than the honest quantum value which is approximately 0.8944. However, the compiler is still outputs a non-trivial game, for instance, when given as input the GHZ game, because the latter has perfect completeness.

12 Soundness Analysis of the $(|C|, 1)$ compiler

The proof is analogous to that of the $(1, 1)$ compiler with some crucial differences and, thus, we skip the intuition and details for parts that are essentially unchanged.

12.1 The Reduction

The procedure for TruthTable generation is the same as in Section 10—except that one samples the context, and more importantly we consider the encrypted answer as well when conditioning, i.e. instead of writing $p_{q'\tau}$, we now use $p_{C,\mathbf{a},\tau}$; we have to add the $\mathbf{a}$ dependence more explicitly because of the following.

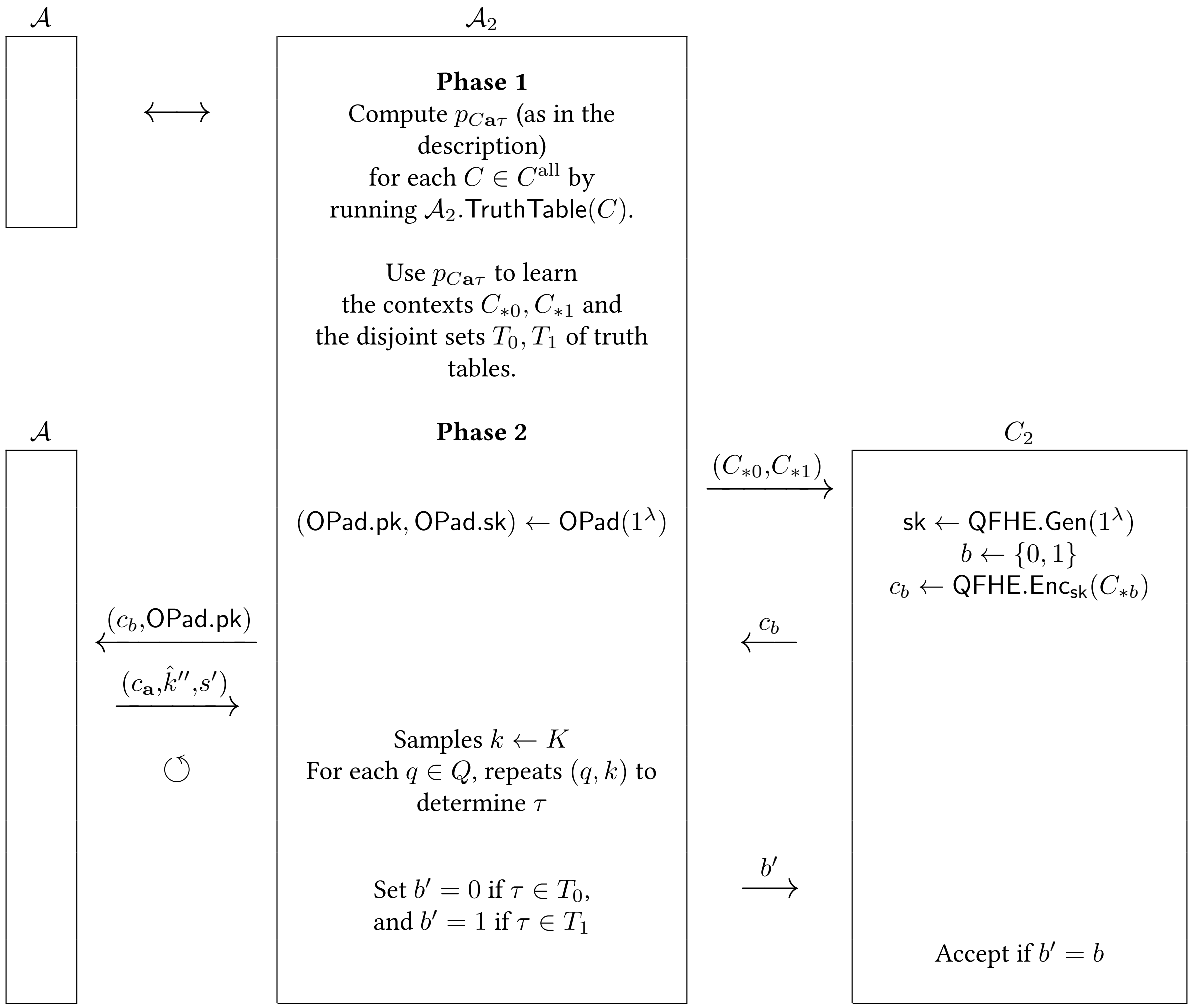
The analogue of Lemma 31 is slightly different. Instead of having the adversary be consistent (in the sense that the encrypted answers and the clear answers are always the same by construction), we require the complementary property—the adversary always ensures that the encrypted answer satisfies the predicate, but may not be consistent. Since we now actually rely on the encrypted answers to check whether or not the predicate is satisfied in the analysis, we cannot simply drop the encrypted answer.

Here is the "TruthTable" algorithm and the actual reduction $\mathcal{A}_2$.

Algorithm 16 Analogue of Algorithm 10, except that it explicitly outputs $\mathbf{a}$, in addition to τ.

$\mathcal{A}$		$\mathcal{A}_2.\mathsf{TruthTable}(C)$
		$\mathsf{sk} \leftarrow \mathsf{QFHE.Gen}(1^\lambda)$ $c \leftarrow \mathsf{QFHE.Enc}(C)$ $(\mathsf{OPad.pk}, \mathsf{OPad.sk}) \leftarrow \mathsf{OPad}(1^\lambda)$
	$\xleftarrow{(c, \mathsf{OPad.pk})}$	
	$\xrightarrow{(c_{\mathbf{a}}, \hat{k}'', s')}$	
	$\circlearrowleft$	Samples $k \leftarrow K$ For each $q \in Q$, ask (q, k), receive a and rewind until τ is fully specified.
		Outputs $\mathbf{a} := \mathsf{QFHE.Dec}_{\mathsf{sk}}(\mathbf{a})$ and τ.

Algorithm 17 The algorithm $\mathcal{A}_2$ uses the adversary $\mathcal{A}$ for the compiled contextuality game G', to break the 2-IND security game for the QFHE scheme.



- The definition of $C_{*0}, C_{*1} \in C^{\text{all}}$ is now "indices" such that

$$\sum_{\tau} \left| \sum_{\mathbf{a}} p_{C_{*0}\mathbf{a}\tau} - \sum_{\mathbf{a}} p_{C_{*1}\mathbf{a}\tau} \right| \geq \eta$$

  where η is non-negligible.

- T_0 and T_1 are defined as $\tau \in T_0$ if $\sum_{\mathbf{a}} p_{C_{*0}\mathbf{a}\tau} \geq \sum_{\mathbf{a}} p_{C_{*1}\mathbf{a}\tau}$ and $\tau \in T_1$ otherwise, i.e. $\sum_{\mathbf{a}} p_{C_{*0}\mathbf{a}\tau} < \sum_{\mathbf{a}} p_{C_{*1}\mathbf{a}\tau}$.

12.2 Proof Strategy

Lemma 35 ([Analogue of Lemma 30] Uniformly random k is equivalent to the correct k)**.** *Let B_0 (resp. B_1) be a PPT algorithm that takes $q' \in Q$ as an input, interacts with $\mathcal{A}$ and outputs a bit, as described in Algorithm 12. Then there is a negligible function* negl *such that* $|\Pr[0 \leftarrow \langle B_0, \mathcal{A}\rangle] - \Pr[0 \leftarrow \langle B_1, \mathcal{A}\rangle]| \leq \mathsf{negl}$.

We are finally ready to write the first step which is conceptually different from the previous analysis. It is the analogue of Lemma 31—except that:

Algorithm 18 Whether a PPT adversary $\mathcal{A}$ for the compiled contextuality game G' is used with the correct k or a uniformly random k, it makes no difference, if all algorithms involved are PPT.

$\mathcal{A}$		$B_0(C)$ (resp. $B_1(C)$)
		$\mathsf{sk} \leftarrow \mathsf{QFHE.Gen}(1^\lambda)$ $c \leftarrow \mathsf{QFHE.Enc}_{\mathsf{sk}}(C)$ $(\mathsf{OPad.pk}, \mathsf{OPad.sk}) \leftarrow \mathsf{OPad}(1^\lambda)$
	$\xleftarrow{(c,\mathsf{OPad.pk})}$	
	$\xrightarrow{(c_{\mathbf{a}},\hat{k}'',s')}$	
		B_0 computes $k' := \mathsf{OPad.Dec}(\mathsf{OPad.sk}, s')$ and uses the secret key sk to compute k'' and then finds the k satisfying $U_k = U_{k''}U_{k'}$. (resp. B_1 samples a uniform $k \leftarrow K$).
	(q,k) $\circlearrowleft$	Potentially rewinds $\mathcal{A}$ to this step and queries with (q, k) for arbitrary $q \in Q$. Runs an arbitrary procedure to compute a bit b'.

- Earlier, we obtained the bound by essentially treating $\mathcal{A}$ as being *consistent* (with the answers under the encryption and those in the clear) where by virtue of being consistent, $\mathcal{A}$ could not be *feasible* i.e. $\mathcal{A}$ could not satisfy all the predicates.
- Now, we treat $\mathcal{A}$ as being *feasible* (satisfies all predicates) , but by virtue of being feasible, it cannot be *consistent* (can't have the encrypted answers be consistent with a global truth assignment).

Lemma 36 ([Analogue of Lemma 31] Feasibility only helps)**.** *Let*

- G' *be the compiled game as in Theorem 34, let* $\mathcal{C}$ *be the challenger in* G'*, and let*
- $\mathcal{C}_{k\leftarrow K}$ *be exactly the same as the challenger* $\mathcal{C}$ *except that it samples a uniformly random* k *instead of computing it correctly,*
- $\mathcal{A}$ *be any PPT algorithm that is designed to play the complied contextuality game* G' *and makes* $\mathcal{C}_{k\leftarrow K}$ *accept with probability* p*, i.e.* $\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C}_{k\leftarrow K}\rangle] = p$ *and denote by*
- $p_{C\mathbf{a}\tau}$ *be the probability that on being asked the context* C*, the answers given are* $\mathbf{a}$ *and the truth table produced is* τ *(as defined in Section 12.1)*
- $p'_{C\mathbf{a}\tau}$ *be a probability distribution derived from* $p_{C\mathbf{a}\tau}$ *to be such that* p' *is always feasible and whenever* p *has support on* $C, \mathbf{a}$ *that satisfy the corresponding predicate,* p' *and* p *agree. More precisely, for any* $(C, \mathbf{a}, \tau)$ *satisfying* $p_{C\mathbf{a}\tau} > 0$ *it holds that*
  - *(*p' *matches* p *exactly when* p *is feasible) if* $p_{C\mathbf{a}\tau} > 0 \wedge \mathsf{pred}(\mathbf{a}[C], C) = 1$*,*
    - $p'_{C\mathbf{a}\tau} = p_{C\mathbf{a}\tau}$ *and*
  - *(*p' *is always feasible) if* $\mathsf{pred}(\mathbf{a}[C], C) = 0$*,*
    - $p'_{C\mathbf{a}\tau} = 0$

– *(when p is infeasible, p' preserves probabilities but makes it feasible) if $p_{C\mathbf{a}\tau} > 0 \wedge \mathsf{pred}(\mathbf{a}[C], C) = 0$, there is an*[30] $\mathbf{a}'$ *such that*

  * $p'_{C\mathbf{a}'\tau} = p_{C\mathbf{a}\tau}$.

Then

$$p \leq \sum_{C\mathbf{a}\tau} p'_{C\mathbf{a}\tau} \Pr(C) \frac{1}{|C|} \sum_{q \in C} \delta_{\mathbf{a}[q],\tau(q)}$$

where $\Pr(C)$ denotes the probability with which $\mathcal{C}$ samples the context C.

The analogue of Lemma 32 should go through with almost no changes.

Lemma 37 ([Analogue of Lemma 32] Precision is not an issue)**.** *Exactly as Lemma 32 except that $\mathcal{C}$ is the challenger for the game compiled using the $(|C|, 1)$ compiler.*

As before, we prove the contrapositive of the soundness condition.

Theorem 38 (Soundness condition restated from Theorem 34)**.** *Suppose*

- *$\mathcal{A}$ is any PPT algorithm that wins with probability*

$$\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C} \rangle] \geq 1 - \mathsf{const}_1 + \epsilon$$

  for some non-negligible function ϵ where $\mathsf{const}_1 = \min_{C \in C^{\mathrm{all}}} \Pr(C)/|C| = O(1)$, and

- *the OPad used is secure, then*

there is a PPT algorithm $\mathcal{A}_{2,\epsilon}$ that wins the 2-IND security game of the QFHE scheme with probability

$$\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}_{2,\epsilon}, C_2 \rangle] \geq \frac{1}{2} + \mathsf{nonnegl},$$

where $\mathsf{nonnegl}$ is a non-negligible function that depends on ϵ.

In Section 12.3, we prove Theorem 38 assuming Lemmas 35, 31 and 32. In Section 12.4 we prove the lemmas.

12.3 Proof assuming the lemmas (Step 1 of 2)

We introduce some notation. Recall the definition of $p_{C\mathbf{a}\tau}$ from Section 12.1 and of $p'_{C\mathbf{a}\tau}$ from Lemma 36.

- We use the notation $p_{\mathbf{a}\tau|C} := p_{C\mathbf{a}\tau}$ since C was given as input to the procedure $\mathcal{C}_2 \cdot \mathsf{TruthTable}$ and it produced outputs $\mathbf{a}, \tau$. Similarly define $p'_{\mathbf{a}\tau|C} := p'_{C\mathbf{a}\tau}$.

- We also use

$$p_{\mathbf{a}\tau|C} = p_{\mathbf{a}|\tau C} \cdot p_{\tau|C} \text{ and } p'_{\mathbf{a}\tau|C} = p'_{\mathbf{a}|\tau C} \cdot p'_{\tau|C} \tag{38}$$

  to denote conditionals.[31]

- Finally, we use the convention of denoting marginals by dropping the corresponding index, i.e.

$$p_{\tau|C} := \sum_{\mathbf{a}} p_{\mathbf{a}\tau|C} \text{ and } p'_{\tau|C} := \sum_{\mathbf{a}} p'_{\mathbf{a}\tau|C}.$$

- Note that by definition of $p'_{\mathbf{a}\tau|C}$, it holds that

$$p'_{\tau|C} = p_{\tau|C}. \tag{39}$$

[30]The first property implies $\mathsf{pred}(\mathbf{a}'[C], C) = 1$.

[31]Using $p(a, b|c) = p(a|b, c) \cdot p(b|c) = \frac{p(a,b,c)}{p(b,c)} \cdot \frac{p(b,c)}{p(c)}$.

Proof. From Lemma 35, one concludes that

$$|[\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C} \rangle] - \Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C}_{k \leftarrow K} \rangle]| \le \mathsf{negl}.$$

Recall the definition of $p_{C\mathbf{a}\tau}$ from Section 12.1. Using Lemma 31, one can write

$$\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C} \rangle] \le \sum_{C\mathbf{a}\tau} p'_{\mathbf{a}\tau|C} \Pr(C) \frac{1}{|C|} \sum_{q \in C} \delta_{\mathbf{a}[q],\tau[q]} + \mathsf{negl} \tag{40}$$

$$= \sum_{C\tau} p_{\tau|C} \sum_{\mathbf{a}} p'_{\mathbf{a}|\tau C} \Pr(C) \frac{1}{|C|} \sum_{q \in C} \delta_{\mathbf{a}[q],\tau[q]} + \mathsf{negl} \qquad \text{using (38) \& (39)} \tag{41}$$

Observe also that there exist $C_{*0} \neq C_{*1}$ such that

$$\sum_{\tau} \left| p_{\tau|C_{*0}} - p_{\tau|C_{*1}} \right| \ge \eta \tag{42}$$

for some non-negligible function η. This is a consequence of the assumption that

$$\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C} \rangle] \ge 1 - \mathsf{const}_1 + \epsilon \tag{43}$$

for some non-negligible function ϵ. To see this, proceed by contradiction: Suppose that for all $C_{*0} \neq C_{*1}$, it is the case that

$$\sum_{\tau} |p_{\tau|C_{*0}} - p_{\tau|C_{*1}}| \le \mathsf{negl} \tag{44}$$

then, setting $p_\tau := p_{\tau|C_{*0}}$ (for instance), it holds that (using Equation (41))

$$\begin{aligned}
\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C} \rangle] &\le \sum_{\tau} p_\tau \sum_{C} \Pr(C) \sum_{\mathbf{a}} p'_{\mathbf{a}|\tau C} \sum_{q \in C} \frac{\delta_{\mathbf{a}[q],\tau[q]}}{|C|} + \mathsf{negl}' \\
&\le \sum_{\tau} p_\tau \left(\underbrace{\sum_{C \neq C_\tau} \Pr(C) \cdot 1}_{\text{Term 1}} + \underbrace{\Pr(C_\tau) \cdot \left(1 - \frac{1}{|C|}\right)}_{\text{Term 2}} \right) + \mathsf{negl}' \\
&\le \sum_{\tau} p_\tau \left(1 - \frac{\Pr(C_\tau)}{|C|}\right) + \mathsf{negl}' \le 1 - \frac{\min_{C \in C^{\mathrm{all}}} \Pr(C)}{|C|} + \mathsf{negl}' = 1 - \mathsf{const}_1 + \mathsf{negl}'
\end{aligned} \tag{45}$$

where the first line uses Equation (44) to substitute $p_{\tau|C}$ with p_τ at the cost a negligible factor but the second line needs some explanation. Observe that (a) for each truth table τ, there is a context $C_\tau \in C^{\mathrm{all}}$ such that $\mathsf{pred}(\tau[C], C) = 0$ because $\mathrm{valNC} < 1$. Observe also that (b) by assumption on $p'_{\mathbf{a}\tau|C}$ (recall Lemma 36), all $\mathbf{a}$ with non-zero weight are feasible, i.e. $\mathsf{pred}(\mathbf{a}[C_\tau], C_\tau) = 1$ for any $p'_{\mathbf{a}|\tau C} > 0$. This implies that there is at least one question $q \in C_\tau$ such that $\mathbf{a}[q] \neq \tau[q]$ (i.e. $\delta_{\mathbf{a}[q],\tau[q]} = 0$)—else τ would also have satisfied the predicate on C_τ which it does not. Using (a) and (b), in Term 1, we simply upper bound the remaining sum by 1 while in term 2, we upper bound the sum by concluding that for at least one question in C_τ, the delta function vanishes. The last inequality follows by setting term 1 to be $1 - \Pr(C_\tau)$ and rearranging and minimising. Now, since Equation (45) contradicts Equation (43) we conclude that our assumption Equation (44) must be false, establishing Equation (42).

The remaining analysis goes through almost unchanged from the $(1, 1)$ compiler case.

$$\begin{aligned}
\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}_2, C_2 \rangle] =& \frac{1}{2} \cdot \sum_{\tau \in T_0} \Pr[\mathcal{A} \text{ outputs } \tau | C_{*0} \text{ was encrypted}] + \\
& \frac{1}{2} \cdot \sum_{\tau \in T_1} \Pr[\mathcal{A} \text{ outputs } \tau | C_{*1} \text{ was encrypted}] \\
=& \frac{1}{2} + \frac{1}{2} \sum_{\tau \in T_0} (p_{\tau|C_{*0}} - p_{\tau|C_{*1}}) \\
=& \frac{1}{2} + \frac{1}{4} \sum_{\tau} \left| p_{\tau|C_{*0}} - p_{\tau|C_{*1}} \right| = \frac{1}{2} + \frac{\eta}{4}.
\end{aligned}$$

This, together with Lemma 37, yields a contradiction with the security of the QFHE scheme. We therefore conclude that Equation (43) is false, which means $\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C} \rangle] \leq 1 - \mathsf{const}_1 + \mathsf{negl}$. $\square$

12.4 Proof of the lemmas (Step 2 of 2)

The proofs of Lemma 35 and Lemma 37 are analogous to those of Lemma 30 and Lemma 32. Here, we prove Lemma 36.

Algorithm 19 $\mathcal{A}'$ uses $\mathcal{A}$ (a PPT algorithm—crucial because it is rewound), to play the compiled contextuality game G'. Its winning probability upper bounds that of $\mathcal{A}$ and can be computed in terms of $p_{\mathbf{a}\tau|C}$ for $\mathcal{A}$.

$\mathcal{A}$		$\mathcal{A}'$		$\mathcal{C}_{k \leftarrow K} =: \mathcal{C}'$
	$\xleftarrow{(c,\mathsf{OPad.pk})}$		$\xleftarrow{(c,\mathsf{OPad.pk})}$	Recall that $c \leftarrow \mathsf{QFHE.Enc}_{\mathsf{sk}}(C)$.
	$\xrightarrow{(c_{\mathbf{a}},\hat{k}'',s')}$			
	$(\tilde{q},\tilde{k})$ $\circlearrowleft$	$\tilde{k} \leftarrow K$ Rewinds $\mathcal{A}'$ to learn the truth table τ. Under the QFHE encryption, with probability $p_{\mathbf{a}|\tau C}$ outputs $c_{\mathbf{a}'} \leftarrow \mathsf{QFHE.Eval}_{\mathbf{a}}(c)$ where $\mathbf{a}' = \mathbf{a}$ if $\mathsf{pred}(\mathbf{a}[C], C) = 1$, otherwise $\mathbf{a}' = \mathbf{a}_C$.		
			$\xrightarrow{(c_{\mathbf{a}'},\hat{k}'',s')}$	
			$\xleftarrow{(q,k)}$	
			$\xrightarrow{a:=\tau(q)}$	(Recall:) Accept if both (1) and (2) hold: (1) $\mathrm{pred}(\mathbf{a}', C) = 1$ and (2) $\mathbf{a}'[q] = a$.

Proof of Lemma 36. Let $p_{\mathbf{a}\tau|c}$ be as in Section 12.3 and recall that $p_{\mathbf{a}\tau|C} = p_{\mathbf{a}|\tau C} p_{\tau|C}$. For each $C \in C^{\text{all}}$, denote by $\mathbf{a}_C$ answers such that $\mathsf{pred}(\mathbf{a}_C, C) = 1$. Consider the adversary $\mathcal{A}'$ as in Algorithm 19. Since this is just a way to compute an upper bound, the running time of $\mathcal{A}'$ does not matter. We show that

$$\begin{aligned} \Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C}' \rangle] &\leq \Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}', \mathcal{C}' \rangle] \\ &= \sum_C \Pr(C) \sum_\tau p_{\tau|C} \sum_{\mathbf{a}'} p'_{\mathbf{a}'|\tau C} \sum_{q \in C} \delta_{\mathbf{a}'[q], \tau[q]} \end{aligned}$$

where the first line holds because, by construction, $\mathcal{A}'$ can do no worse than $\mathcal{A}$. More specifically, $\mathcal{A}'$ behaves exactly like $\mathcal{A}$ when $\mathbf{a}$ satisfies the predicate, i.e. $\mathbf{a}' = \mathbf{a}$, and when $\mathbf{a}$ fails the predicate, $\mathcal{A}'$ only increases its probability of success by responding with $\mathbf{a}' \neq \mathbf{a}$ such that $\mathsf{pred}(\mathbf{a}', C) = 1$. The second line is straightforward. Denote by $p'_{\tau|C}$ the probability with which $\mathcal{A}'$ responds with τ on being asked C. Similarly, let $p'_{\mathbf{a}'|\tau C}$ be the probability that $\mathcal{A}'$ responds with $\mathbf{a}'$ given τ and C. Then, the challenger $\mathcal{C}'$ asks a context C with probability $\Pr(C)$, to which the prover $\mathcal{A}'$ responds with τ with probability $p'_{\tau|C} = p_{\tau|C}$ and given τC, it responds with $\mathbf{a}'$ with probability $p'_{\mathbf{a}'|\tau C}$. It follows that $p'_{\mathbf{a}'\tau|C} := p'_{\mathbf{a}'|\tau C} p'_{\tau|C}$ satisfies the asserted properties: it has no support over $(\mathbf{a}', C)$ that are not feasible (don't satisfy the predicate for the corresponding context C) and whenever $(\mathbf{a}, C)$ is feasible, $\mathbf{a}' = \mathbf{a}$ by construction. $\square$

13 Construction of the $(|C|-1,1)$ Compiler

We define the compiler formally.

Algorithm 20 Game G' produced by the $(1,1)$-compiler for any contextuality game G with contexts of size two.

Honest Prover ($\mathcal{A}$)		Challenger ($\mathcal{C}$)
		$\mathsf{sk} \leftarrow \mathsf{QFHE.Gen}(1^\lambda)$ $C \leftarrow \mathcal{D}$ $q_{\mathsf{skip}} \leftarrow C$ $C' := C \backslash q_{\mathsf{skip}}$ $c \leftarrow \mathsf{QFHE.Enc}_{\mathsf{sk}}(C')$ $(\mathsf{OPad.pk}, \mathsf{OPad.sk}) \leftarrow \mathsf{OPad}(1^\lambda)$
	$\xleftarrow{(c, \mathsf{OPad.pk})}$	
Under the QFHE encryption, measures $\{O_{q'}\}_{q' \in C'}$ and obtains encrypted answers $c_{\mathbf{a}'}$ (where $\mathbf{a}'$ are the answers, indexed by C') and the post-measurement state $(U_{k''} \|\psi_{C'\mathbf{a}'}\rangle, \hat{k}'')$. Applies an oblivious $\mathbf{U}$-pad to this state to obtain $(U_{k'}U_{k''} \|\psi_{C'\mathbf{a}'}\rangle, s') \leftarrow$ $\mathsf{OPad.Enc}(\mathsf{OPad.pk}, U_{k''} \|\psi_{q'a'}\rangle)$.		
	$\xrightarrow{(c_{\mathbf{a}'}, \hat{k}'', s')}$	
		Using the secret keys sk, $\mathsf{OPad.sk}$, finds the k such that $U_k = U_{k''}U_{k'}$, samples $q \leftarrow C$
	$\xleftarrow{(q,k)}$	
Measures $U_k O_q U_k^\dagger$ and obtains a		
	$\xrightarrow{a}$	
		Computes $\mathbf{a}' = \mathsf{Dec}_{\mathsf{sk}}(c_{\mathbf{a}'})$. If $q \in C'$, accept if $a = \mathbf{a}'[q]$ If $q \notin C'$ (i.e. $q = q_{\mathsf{skip}}$), accept if $\mathrm{pred}(\mathbf{a}' \cup a, C) = 1$ where $\mathbf{a}' \cup a$ denotes answers indexed by C.

13.1 Compiler Guarantees

The compiler satisfies the following. Without loss of generality, we restrict to contextuality games where all contexts have the same size (see Remark 13).

Theorem 39 (Guarantees of the $(|C|-1,1)$ compiled contextuality game G')**.** *Suppose* QFHE *and* OPad *are secure (as in Definitions 5 and 19), and compatible (as in Definition 22). Let* G *be any contextuality game with* $\mathrm{valNC} < 1$ *where all contexts are of the same size (i.e.* $|C| = |C'|$ *for all* $C, C' \in C^{\mathrm{all}}$*). Let* G'_λ *be the compiled game produced by Algorithm 20 on input* G *and a security parameter* λ*. Then, the following holds.*

- *(Completeness) There is a negligible function* negl*, such that, for all* $\lambda \in \mathbb{N}$*, the honest QPT prover from Algorithm 20 wins* G'_λ *with probability at least*

$$c(\lambda) := 1 - \frac{1}{|C|} + \frac{\text{valQu}}{|C|} - \mathsf{negl}(\lambda).$$

- *(Soundness) For every PPT adversary* $\mathcal{A}$*, there is a negligible function* negl' *such that, for all* $\lambda \in \mathbb{N}$*, the probability that* $\mathcal{A}$ *wins* G'_λ *is at most*

$$s(\lambda) := 1 - \frac{1}{|C|} + \frac{\text{valNC}}{|C|} + \mathsf{negl}'(\lambda)\,,$$

Furthermore, G'_λ *is faithful to* G *(as in Definition 18) with parameters* $s(\lambda)$ *and* $c(\lambda)$*.*

Completeness is straightforward to verify. The proof of faithfulness is analogous to that of Theorem 24. We prove soundness in Section 14.

14 Soundness Analysis of the $(|C| - 1, 1)$ compiler

14.1 The Reduction

Denote by $\mathsf{C}'^{\mathsf{all}} := \{C \backslash q_{\mathsf{skip}}\}_{C \in C^{\mathrm{all}}, q_{\mathsf{skip}} \in C}$ the set consisting of contexts with exactly one question removed.

The reduction is very similar to that in Section 10.2, so we do not repeat the accompanying high-level explanations. Let $\mathcal{A}_2 \cdot \mathsf{TruthTable}$ be as defined in Algorithm 22. Define $p_{C'\tau}$ to be the probability that the procedure $\mathcal{A}_2 \cdot \mathsf{TruthTable}(C')$ outputs τ, where $C' \in \mathsf{C}'^{\mathsf{all}}$ is a context with exactly one question removed.

Algorithm 21 The procedure $\mathcal{A}_2.\mathsf{TruthTable}$ takes as input a context with one question excluded, $C' = C \backslash q_{\mathsf{skip}}$. It produces a truth table τ corresponding to it. Note that this is a randomised procedure (depends on the QFHE encryption procedure) so for the same C' the procedure may output different τs. The goal is to learn the probabilities of different τs appearing for each question C'.

$\mathcal{A}$		$\mathcal{A}_2.\mathsf{TruthTable}(C')$
		$\mathsf{sk} \leftarrow \mathsf{QFHE.Gen}(1^\lambda)$ $c \leftarrow \mathsf{QFHE.Enc}(C')$ $(\mathsf{OPad.pk}, \mathsf{OPad.sk}) \leftarrow \mathsf{OPad}(1^\lambda)$
	$\xleftarrow{(c, \mathsf{OPad.pk})}$	
	$\xrightarrow{(c_{\mathbf{a}'}, \hat{k}'', s')}$	
	$\circlearrowleft$	Samples $k \leftarrow K$ For each $q \in Q$, ask (q, k), receive a and rewind until τ is fully specified.
		Outputs τ

14.2 Proof Strategy

Lemma 40 (Uniformly random k is equivalent to the correct k)**.** *Let* B_0 *(resp.* B_1*) be a PPT algorithm that takes* $C' \in \mathsf{C}'^{\mathsf{all}}$ *as an input, interacts with* $\mathcal{A}$ *and outputs a bit, as described in Algorithm 23. Then, there is a negligible function* negl *such that* $|\Pr[0 \leftarrow \langle B_0, \mathcal{A}\rangle] - \Pr[0 \leftarrow \langle B_1, \mathcal{A}\rangle]| \leq \mathsf{negl}$.

The following we will check carefully at the end. It says that one can treat $\mathcal{A}$ as though it is consistent.

Algorithm 22 The algorithm $\mathcal{A}_2$ uses the adversary $\mathcal{A}$ for the compiled contextuality game G', to break the 2-IND security game for the QFHE scheme.

$\mathcal{A}$		$\mathcal{A}_2$		
	$\longleftrightarrow$	**Phase 1** Compute $p_{C'\tau}$ (as in the description) for each C' by running $\mathcal{A}_2.\mathsf{TruthTable}(C')$. Use $p_{C'\tau}$ to learn the questions C'_{*0}, C'_{*1} and the disjoint sets T_0, T_1 of truth tables.		
$\mathcal{A}$		**Phase 2**		C_2
			$\xrightarrow{(C'_{*0},C'_{*1})}$	
		$(\mathsf{OPad.pk}, \mathsf{OPad.sk}) \leftarrow \mathsf{OPad}(1^\lambda)$		$\mathsf{sk} \leftarrow \mathsf{QFHE.Gen}(1^\lambda)$ $b \leftarrow \{0,1\}$ $c_b \leftarrow \mathsf{QFHE.Enc}_{\mathsf{sk}}(q_{*b})$
	$\xleftarrow{(c_b,\mathsf{OPad.pk})}$		$\xleftarrow{c_b}$	
	$\xrightarrow{(c_{\mathbf{a}'},\hat{k}'',s')}$			
	$\circlearrowleft$	Samples $k \leftarrow K$ For each $q \in Q$, repeats (q, k) to determine τ		
		Set $b' = 0$ if $\tau \in T_0$, and $b' = 1$ if $\tau \in T_1$	$\xrightarrow{b'}$	
				Accept if $b' = b$

Algorithm 23 Whether a PPT adversary $\mathcal{A}$ for the compiled contextuality game G' is used with the correct k or a uniformly random k, it makes no difference, if all algorithms involved are PPT.

$\mathcal{A}$		$B_0(C')$ (resp. $B_1(C')$)
		$\mathsf{sk} \leftarrow \mathsf{QFHE.Gen}(1^\lambda)$ $c \leftarrow \mathsf{QFHE.Enc}_{\mathsf{sk}}(C')$ $(\mathsf{OPad.pk}, \mathsf{OPad.sk}) \leftarrow \mathsf{OPad}(1^\lambda)$
	$\xleftarrow{(c,\mathsf{OPad.pk})}$	
	$\xrightarrow{(c_{\mathbf{a}'},\hat{k}'',s')}$	
		B_0 computes $k' := \mathsf{OPad.Dec}(\mathsf{OPad.sk}, s')$ and uses the secret key sk to compute k'' and then finds the k satisfying $U_k = U_{k''}U_{k'}$. (resp. B_1 samples a uniform $k \leftarrow K$).
	(q,k) $\circlearrowleft$	Potentially rewinds $\mathcal{A}$ to this step and queries with (q, k) for arbitrary $q \in Q$. Runs an arbitrary procedure to compute a bit b'.

Lemma 41 (Consistency only helps). *Let*

- $\mathcal{C}_{k\leftarrow K}$ *be exactly the same as the challenger* $\mathcal{C}$ *for* G' *except that it samples* $k \leftarrow K$ *uniformly, instead of computing it correctly, let*
- $\mathcal{A}$ *be any PPT algorithm that plays* G' *and* $\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C}_{k\leftarrow K}\rangle] = p$ *and denote by*
- $p_{C'\tau}$ *be as described above.*

Then

$$p \leq \left(1 - \frac{1}{|C|}\right) + \sum_{C \in C^{\mathrm{all}}} \Pr(C) \sum_{q_{\mathsf{skip}} \in C} \frac{1}{|C|} \sum_{\tau} p_{C',\tau} \mathsf{pred}(\tau[C], C)$$

where $C' = C \backslash q_{\mathsf{skip}}$*, and* $\Pr(C)$ *denotes the probability with which* $\mathcal{C}$ *samples the context* C*.*

The first term captures the probability that the consistency test passes and the second one captures the probability that the predicate test passes.

Lemma 42. *Exactly the same as Lemma 32 except that*

- $\mathcal{C}$ *is the challenger for the game produced by the* $(|C| - 1, 1)$ *compiler,*
- *instead of* $p_{q'\tau}$*, use* $p_{C'\tau}$*, and*
- *the construction of* $\mathcal{A}_2$ *is as in Algorithm 22.*

Let $\mathcal{A}$ *be such that* $\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C}\rangle] \geq 1 - \frac{1}{|C|} + \frac{\mathrm{valNC}}{|C|} + \epsilon$ *and let* $\mathcal{A}_{2,\epsilon}$ *be as in Lemma 32. Then, it holds that*

$$\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}_{2,\epsilon}, \mathcal{C}_2\rangle] \geq \Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}_2, \mathcal{C}_2\rangle] - O(\epsilon^3).$$

The following is the contrapositive of the soundness guarantee in Theorem 39.

Theorem 43 (Soundness condition restated from Theorem 39)**.** *Suppose*

- $\mathcal{A}$ *is any PPT algorithm that wins with probability* $\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C} \rangle] \geq 1 - \frac{1}{|C|} + \frac{\text{valNC}}{|C|} + \epsilon$ *for some non-negligible function* ϵ *and*
- *the* OPad *used is secure, then*

there is a PPT algorithm $\mathcal{A}_{2,\epsilon}$ *that wins the* 2*-IND security game of the* QFHE *scheme with probability*

$$\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}_{2,\epsilon}, C_2 \rangle] \geq \frac{1}{2} + \mathsf{nonnegl},$$

where $\mathsf{nonnegl}$ *is a non-negligible function that depends on* ϵ*.*

In Section 14.3, we prove Theorem 43 assuming Lemmas 40, 41 and 42. In Section 14.4, we prove the lemmas.

14.3 Proof assuming the lemmas (Step 1 of 2)

Proof. From Lemma 40, we have that $|\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C} \rangle] - \Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C}_{k\leftarrow K} \rangle]| \leq \mathsf{negl}$. Recall the definition of $p_{C'\tau}$ and use Lemma 41 to write

$$\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C} \rangle] - \mathsf{negl} \leq \left(1 - \frac{1}{|C|}\right) + \sum_{C \in C^{\text{all}}} \Pr(C) \sum_{q_{\mathsf{skip}} \in C} \frac{1}{|C|} \sum_{\tau} p_{C',\tau} \mathsf{pred}(\tau[C], C) \tag{46}$$

where recall that $C' := C \backslash q_{\mathsf{skip}}$. Observe also that there exists $C'_{*0} \neq C'_{*1}$ such that

$$\left\| p_{C'_{*0}} - p_{C'_{*1}} \right\|_1 := \sum_{\tau} \left| p_{C'_{*0}\tau} - p_{C'_{*1}\tau} \right| \geq \eta \tag{47}$$

for some non-negligible function η. This is a consequence of the assumption that

$$\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C} \rangle] \geq 1 - \frac{1}{|C|} + \frac{\text{valNC}}{|C|} + \epsilon \tag{48}$$

for some non-negligible function ϵ. To see this, suppose for contradiction that for all $C'_{*0} \neq C'_{*1}$, it were the case that $\sum_{\tau} |p_{C'_{*0}\tau} - p_{C'_{*1}\tau}| \leq \mathsf{negl}$ for some negligible function, then one could write, using Equation (46) and $p_\tau := p_{C'_{*0}\tau}$

$$\begin{aligned}
\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C} \rangle] \leq & \left(1 - \frac{1}{|C|}\right) + \sum_{C \in C^{\text{all}}} \Pr(C) \cancel{\sum_{q_{\mathsf{skip}} \in C} \frac{1}{|C|}} \sum_{\tau} p_\tau \mathsf{pred}(\tau[C], C) + \mathsf{negl}' \\
= & \left(1 - \frac{1}{|C|}\right) + \sum_{\tau} p_\tau \sum_{C \in C^{\text{all}}} \Pr(C) \mathsf{pred}(\tau[C], C) + \mathsf{negl}' \\
\leq & 1 - \frac{1}{|C|} + \frac{\text{valNC}}{|C|} + \mathsf{negl}'.
\end{aligned}$$

But this contradicts Equation (48) and thus Equation (47) holds for some $C'_{*0} \neq C'_{*1}$ as claimed.

The remaining analysis is the same as the $(1,1)$ case, briefly, note that

$$\begin{aligned}
\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}_2, \mathcal{C}_2 \rangle] = & \frac{1}{2} \cdot \sum_{\tau \in T_0} \Pr[\mathcal{A} \text{ outputs } \tau | C'_{*0} \text{ was encrypted}] + \\
& \frac{1}{2} \cdot \sum_{\tau \in T_1} \Pr[\mathcal{A} \text{ outputs } \tau | C'_{*1} \text{ was encrypted}] \\
= & \frac{1}{2} + \frac{1}{2} \sum_{\tau \in T_0} (p_{C'_{*0}\tau} - p_{C'_{*1}\tau}) \\
\geq & \frac{1}{2} + \frac{\eta}{4}.
\end{aligned}$$

Since η is non-negligible, existence of a PPT $\mathcal{A}$ satisfying Equation (48) breaks the security of the underlying QFHE scheme. The precision issue is handled by invoking Lemma 42. This completes the proof. □

14.4 Proof of the lemmas (Step 2 of 2)

We only prove Lemma 41. The proofs of Lemma 40 and Lemma 42 are analogous to those of Lemma 30 and Lemma 32, respectively.

Proof of Lemma 41. Consider the adversary $\mathcal{A}'$ in Algorithm 24 that uses $\mathcal{A}$ to interact with $\mathcal{C}_{k\leftarrow K} =: \mathcal{C}'$ (not to be confused with C' which denotes a context with exactly one question removed). We show that

Algorithm 24 $\mathcal{A}'$ uses $\mathcal{A}$ (a PPT algorithm), to play the compiled contextuality game G'. Its winning probability upper bounds that of $\mathcal{A}$ and can be computed in terms of $p_{C'\tau}$ for $\mathcal{A}$.

$\mathcal{A}$		$\mathcal{A}'$		$\mathcal{C}_{k\leftarrow K} =: \mathcal{C}'$
	$\xleftarrow{(c,\mathsf{OPad.pk})}$		$\xleftarrow{(c,\mathsf{OPad.pk})}$	Recall: $c \leftarrow \mathsf{Enc}_{\mathsf{sk}}(C')$ for $C' = C\backslash q_{\mathsf{skip}}$.
	$\xrightarrow{(c_{\mathbf{a}'},\hat{k}'',s')}$			
	$(\tilde{q},\tilde{k})$	$\tilde{k} \leftarrow K$ Rewinds $\mathcal{A}'$ to learn the truth table τ. Evaluate $c_{\tau[C']} \leftarrow \mathsf{QFHE.Eval}_\tau(c)$.		
			$\xrightarrow{(c_{\tau(q')},\hat{k}'',s')}$	
			$\xleftarrow{(q,k)}$	
				Recall: $\mathbf{a}' = \mathsf{Dec}_{\mathsf{sk}}(c_{\tau[C']}) = \tau[C']$
			$\xrightarrow{a:=\tau(q)}$	If $q \in C'$, accept if $a = \mathbf{a}'[q]$ If $q \notin C'$ (i.e. $q = q_{\mathsf{skip}}$), accept if $\mathrm{pred}(\mathbf{a}' \cup a, C) = 1$ where $\mathbf{a}' \cup a$ denotes answers indexed by C.

$$\Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}, \mathcal{C}' \rangle] \leq \Pr[\mathsf{accept} \leftarrow \langle \mathcal{A}', \mathcal{C}' \rangle] \tag{49}$$

$$= \left(1 - \frac{1}{|C|}\right) + \sum_{C \in C^{\mathrm{all}}} \Pr(C) \sum_{q_{\mathsf{skip}} \in C} \frac{1}{|C|} \sum_{\tau} p_{C',\tau} \mathsf{pred}(\tau[C], C) \tag{50}$$

where recall that $C' = C\backslash q_{\mathsf{skip}}$.

Note that for any given C', the challenger asks any specific q with probability $1/|C|$, which in particular means that $q = q_{\mathsf{skip}}$ with probability $1/|C|$ and $q \neq q_{\mathsf{skip}}$ with probability $1 - 1/|C|$.

Let's derive Equation (49). Consider the interactions $\langle \mathcal{A}, \mathcal{C}' \rangle$ and $\langle \mathcal{A}', \mathcal{C}' \rangle$. Conditioned on C', there are two cases: (1) $\mathcal{A}$ is consistent, in which case, the answers given by $\mathcal{A}'$ and $\mathcal{A}$ are identical, or (2) $\mathcal{A}$ is inconsistent in which case $\mathcal{A}$ fails with probability *at least* $1/|C|$ (because the challenger spots the inconsistency with probability at least $1/|C|$), while $\mathcal{A}'$ fails with probability *at most* $1/|C|$ (because it at most (potentially) fails the predicate evaluation, which happens with probability exactly $1/|C|$).

As for Equation (50), it follows because $\mathcal{C}'$ selects a context C with probability $\Pr(C)$, it asks $q \neq q_{\mathsf{skip}}$ with probability $1 - 1/|C|$ which corresponds to doing a consistency test—which $\mathcal{A}'$ passes with probability 1 by construction. Finally, note that $\mathbf{a}' \cup a = \tau[C]$. Now, $\mathcal{C}'$ asks $q = q_{\mathsf{skip}}$ with probability $1/|C|$ and in this case, $\mathcal{A}'$ responds with $\tau[C]$ with probability $p_{C',\tau}$. Thus, its success probability in this case is the weighted average of $\mathsf{pred}(\tau[C], C) = \mathsf{pred}(\mathbf{a}' \cup a, C)$ where the weights are given by $p_{C',\tau}$ (recall that $C' = C\backslash q_{\mathsf{skip}}$). This completes the proof. ∎

References

[ACGH20] Gorjan Alagic, Andrew M Childs, Alex B Grilo, and Shih-Han Hung. Non-interactive classical verification of quantum computation. In *Theory of Cryptography Conference*, pages 153–180. Springer, 2020.

[AMMW22] Yusuf Alnawakhtha, Atul Mantri, Carl A Miller, and Daochen Wang. Lattice-based quantum advantage from rotated measurements. *arXiv preprint arXiv:2210.10143*, 2022.

[AMP23] Alastair A Abbott, Mehdi Mhalla, and Pierre Pocreau. Quantum query complexity of boolean functions under indefinite causal order. *arXiv preprint arXiv:2307.10285*, 2023.

[Ara04] Padmanabhan K Aravind. Quantum mysteries revisited again. *American Journal of Physics*, 72(10):1303–1307, 2004.

[BCG+22] Costantino Budroni, Adán Cabello, Otfried Gühne, Matthias Kleinmann, and Jan-Åke Larsson. Kochen-specker contextuality. *Reviews of Modern Physics*, 94(4):045007, 2022.

[BCM+21] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. *J. ACM*, 68(5), aug 2021.

[Bel64] John S Bell. On the einstein podolsky rosen paradox. *Physics Physique Fizika*, 1(3):195, 1964.

[BGKM+23] 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*, pages 162–191, Cham, 2023. Springer Nature Switzerland.

[BKVV20] Zvika Brakerski, Venkata Koppula, Umesh 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)*, volume 158 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[BMKG13] Costantino Budroni, Tobias Moroder, Matthias Kleinmann, and Otfried Gühne. Bounding temporal quantum correlations. *Physical review letters*, 111(2):020403, 2013.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and Communications Security*, pages 62–73, 1993.

[Bra18] Zvika Brakerski. Quantum fhe (almost) as secure as classical. In *Advances in Cryptology - CRYPTO 2018: 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*, pages 67–95, Berlin, Heidelberg, 2018. Springer-Verlag.

[BRV+19a] Kishor Bharti, Maharshi Ray, Antonios Varvitsiotis, Adán Cabello, and Leong-Chuan Kwek. Local certification of programmable quantum devices of arbitrary high dimensionality. *arXiv preprint arXiv:1911.09448*, 2019.

[BRV+19b] Kishor Bharti, Maharshi Ray, Antonios Varvitsiotis, Naqueeb Ahmad Warsi, Adán Cabello, and Leong-Chuan Kwek. Robust self-testing of quantum systems via noncontextuality inequalities. *Phys. Rev. Lett.*, 122:250403, Jun 2019.

[Cab01] Adán Cabello. Bell's theorem without inequalities and without probabilities for two observers. *Physical review letters*, 86(10):1911, 2001.

[CDPV13] Giulio Chiribella, Giacomo Mauro D'Ariano, Paolo Perinotti, and Benoit Valiron. Quantum computations without definite causal structure. *Physical Review A*, 88(2):022318, 2013.

[CGGX18] Adán Cabello, Mile Gu, Otfried Gühne, and Zhen-Peng Xu. Optimal classical simulation of state-independent quantum contextuality. *Physical review letters*, 120(13):130401, 2018.

[CGJV19] 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.

[CHSH69] John F Clauser, Michael A Horne, Abner Shimony, and Richard A Holt. Proposed experiment to test local hidden-variable theories. *Physical review letters*, 23(15):880, 1969.

[CMM+24] David Cui, Giulio Malavolta, Arthur Mehta, Anand Natarajan, Connor Paddock, Simon Schmidt, Michael Walter, and Tina Zhang. A computational tsirelson's theorem for the value of compiled XOR games. *IACR Cryptol. ePrint Arch.*, page 348, 2024.

[EPR35] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780, May 1935.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, pages 186–194, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, pages 169–178, New York, NY, USA, 2009. Association for Computing Machinery.

[GVW+15] Marissa Giustina, Marijn AM Versteegh, Sören Wengerowsky, Johannes Handsteiner, Armin Hochrainer, Kevin Phelan, Fabian Steinlechner, Johannes Kofler, Jan-Åke Larsson, Carlos Abellán, et al. Significant-loophole-free test of bell's theorem with entangled photons. *Physical review letters*, 115(25):250401, 2015.

[HBD+15] Bas Hensen, Hannes Bernien, Anaïs E Dréau, Andreas Reiserer, Norbert Kalb, Machiel S Blok, Just Ruitenberg, Raymond FL Vermeulen, Raymond N Schouten, Carlos Abellán, et al. Loophole-free bell inequality violation using electron spins separated by 1.3 kilometres. *Nature*, 526(7575):682–686, 2015.

[HKS13] Dennis Hofheinz, Eike Kiltz, and Victor Shoup. Practical chosen ciphertext secure encryption from factoring. *J. Cryptol.*, 26(1):102–118, January 2013.

[HXA+23] Xiao-Min Hu, Yi Xie, Atul Singh Arora, Ming-Zhong Ai, Kishor Bharti, Jie Zhang, Wei Wu, Ping-Xing Chen, Jin-Ming Cui, Bi-Heng Liu, et al. Self-testing of a single quantum system from theory to experiment. *npj Quantum Information*, 9(1):103, 2023.

[JRO+16] Markus Jerger, Yarema Reshitnyk, Markus Oppliger, Anton Potočnik, Mintu Mondal, Andreas Wallraff, Kenneth Goodenough, Stephanie Wehner, Kristinn Juliusson, Nathan K Langford, et al. Contextuality without nonlocality in a superconducting quantum system. *Nature communications*, 7(1):12930, 2016.

[KCBS08] Alexander A Klyachko, M Ali Can, Sinem Binicioğlu, and Alexander S Shumovsky. Simple test for hidden variables in spin-1 systems. *Physical review letters*, 101(2):020403, 2008.

[KGP+11] Matthias Kleinmann, Otfried Guehne, Jose R Portillo, Jan-Åke Larsson, and Adan Cabello. Memory cost of quantum contextuality. *New Journal of Physics*, 13(11):113011, 2011.

[KLVY23] Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Lisa Yang. Quantum advantage from any non-local game. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pages 1617–1628, New York, NY, USA, 2023. Association for Computing Machinery.

[KMCVY22] Gregory D. Kahanamoku-Meyer, Soonwon Choi, Umesh V. Vazirani, and Norman Y. Yao. Classically verifiable quantum advantage from a computational bell test. *Nature Physics*, 18(8):918–924, August 2022.

[KMY24] Gregory D. Kahanamoku-Meyer and Norman Y. Yao. Fast quantum integer multiplication with zero ancillas, 2024.

[KS67] Simon Kochen and Ernst Specker. The problem of hidden variables in quantum mechanics. *J. Math. Mech.*, 17:59–87, 1967.

[LG85] Anthony J Leggett and Anupam Garg. Quantum mechanics versus macroscopic realism: Is the flux there when nobody looks? *Physical Review Letters*, 54(9):857, 1985.

[Liu23] Zheng-Hao Liu. *Exploring Quantum Contextuality with Photons.* Springer Nature, 2023.

[LLS+11] Radek Lapkiewicz, Peizhe Li, Christoph Schaeff, Nathan K Langford, Sven Ramelow, Marcin Wieśniak, and Anton Zeilinger. Experimental non-classicality of an indivisible quantum system. *Nature*, 474(7352):490–493, 2011.

[LMX+23] Zheng-Hao Liu, Hui-Xian Meng, Zhen-Peng Xu, Jie Zhou, Jing-Ling Chen, Jin-Shi Xu, Chuan-Feng Li, Guang-Can Guo, and Adán Cabello. Experimental test of high-dimensional quantum contextuality based on contextuality concentration. *Physical Review Letters*, 130(24):240202, 2023.

[LMZ+18] Florian M Leupold, Maciej Malinowski, Chi Zhang, Vlad Negnevitsky, Adán Cabello, Joseba Alonso, and Jonathan P Home. Sustained state-independent quantum contextual correlations from a single ion. *Physical review letters*, 120(18):180401, 2018.

[LWZ+18] Ming-Han Li, Cheng Wu, Yanbao Zhang, Wen-Zhao Liu, Bing Bai, Yang Liu, Weijun Zhang, Qi Zhao, Hao Li, Zhen Wang, et al. Test of local realism into the past without detection and locality loopholes. *Physical review letters*, 121(8):080404, 2018.

[Mah18] Urmila Mahadev. Classical verification of quantum computations. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–267. IEEE, 2018.

[Mah20] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. *SIAM Journal on Computing*, 52(6):FOCS18–189, 2020.

[Mer90] N David Mermin. Simple unified form for the major no-hidden-variables theorems. *Physical review letters*, 65(27):3373, 1990.

[MZL+18] Maciej Malinowski, Chi Zhang, Florian M Leupold, Adán Cabello, Joseba Alonso, and JP Home. Probing the limits of correlations in an indivisible quantum system. *Physical Review A*, 98(5):050102, 2018.

[NZ23] A. Natarajan and T. Zhang. Bounding the quantum value of compiled nonlocal games: From chsh to bqp verification. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1342–1348, Los Alamitos, CA, USA, nov 2023. IEEE Computer Society.

[OCB12] Ognyan Oreshkov, Fabio Costa, and Časlav Brukner. Quantum correlations with no causal order. *Nature communications*, 3(1):1092, 2012.

[Per90] Asher Peres. Incompatible results of quantum measurements. *Physics Letters A*, 151(3-4):107–108, 1990.

[RBG+17] Wenjamin Rosenfeld, Daniel Burchardt, Robert Garthoff, Kai Redeker, Norbert Ortegel, Markus Rau, and Harald Weinfurter. Event-ready bell test using entangled atoms simultaneously closing detection and locality loopholes. *Physical review letters*, 119(1):010402, 2017.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), sep 2009.

[RUV13] Ben W. Reichardt, Falk Unger, and Umesh Vazirani. A classical leash for a quantum system: command of quantum systems via rigidity of chsh games. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, pages 321–322, New York, NY, USA, 2013. Association for Computing Machinery.

[SMSC+15] Lynden K Shalm, Evan Meyer-Scott, Bradley G Christensen, Peter Bierhorst, Michael A Wayne, Martin J Stevens, Thomas Gerrits, Scott Glancy, Deny R Hamel, Michael S Allman, et al. Strong loophole-free test of local realism. *Physical review letters*, 115(25):250402, 2015.

[Spe60] Ernst Specker. Die logik nicht gleichzeitig entsc heidbarer aussagen. *Dialectica*, 14(2-3):239–246, September 1960.

[Spe05] Robert W Spekkens. Contextuality for preparations, transformations, and unsharp measurements. *Physical Review A*, 71(5):052108, 2005.

[SSA20] Debashis Saha, Rafael Santos, and Remigiusz Augusiak. Sum-of-squares decompositions for a family of noncontextuality inequalities and self-testing of quantum devices. *Quantum*, 4:302, 2020.

[SSK+23] Simon Storz, Josua Schär, Anatoly Kulikov, Paul Magnard, Philipp Kurpiers, Janis Lütolf, Theo Walter, Adrian Copetudo, Kevin Reuer, Abdulkadir Akin, et al. Loophole-free bell inequality violation with superconducting circuits. *Nature*, 617(7960):265–270, 2023.

[UZZ+13] Mark Um, Xiang Zhang, Junhua Zhang, Ye Wang, Shen Yangchao, D-L Deng, Lu-Ming Duan, and Kihwan Kim. Experimental certification of random numbers via quantum contextuality. *Scientific reports*, 3(1):1627, 2013.

[UZZ+20] Mark Um, Qi Zhao, Junhua Zhang, Pengfei Wang, Ye Wang, Mu Qiao, Hongyi Zhou, Xiongfeng Ma, and Kihwan Kim. Randomness expansion secured by quantum contextuality. *Physical Review Applied*, 13(3):034077, 2020.

[WZL+22] Pengfei Wang, Junhua Zhang, Chun-Yang Luan, Mark Um, Ye Wang, Mu Qiao, Tian Xie, Jing-Ning Zhang, Adán Cabello, and Kihwan Kim. Significant loophole-free test of kochen-specker contextuality using two species of atomic ions. *Science Advances*, 8(6):eabk1660, 2022.

[XSBC24] Zhen-Peng Xu, Debashis Saha, Kishor Bharti, and Adán Cabello. Certifying sets of quantum observables with any full-rank state. *Phys. Rev. Lett.*, 132:140201, Apr 2024.

[YZ22] Takashi Yamakawa and Mark Zhandry. Verifiable quantum advantage without structure. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 69–74. IEEE, 2022.

[ZKK+17] Xiang Zhan, Paweł Kurzyński, Dagomir Kaszlikowski, Kunkun Wang, Zhihao Bian, Yongsheng Zhang, and Peng Xue. Experimental detection of information deficit in a photonic contextuality scenario. *Physical Review Letters*, 119(22):220403, 2017.

[ZXX+19] Aonan Zhang, Huichao Xu, Jie Xie, Han Zhang, Brian J Smith, MS Kim, and Lijian Zhang. Experimental test of contextuality in quantum and classical systems. *Physical review letters*, 122(8):080401, 2019.