Compiling Nonlocal Games without Quantum Homomorphic Encryption

Kaniuar Bacho

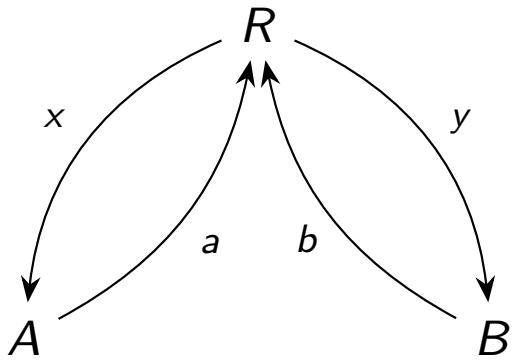
Ruhr University Bochum

December 9, 2024

Outline

Nonlocal Games

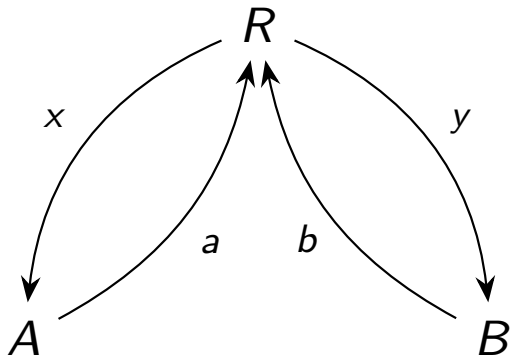
A nonlocal game $\mathcal{G}$ is a classical interaction between a referee R and two cooperating players, Alice and Bob:



- Alice and Bob cannot communicate after receiving the questions x and y.
- They win if $V(a, b, x, y) = 1$ for some predefined verification function V.

Nonlocal Games

A nonlocal game $\mathcal{G}$ is a classical interaction between a referee R and two cooperating players, Alice and Bob:

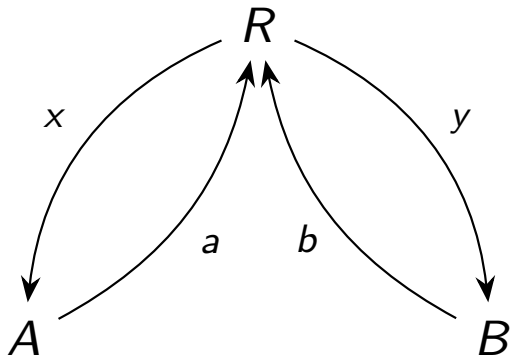


- Alice and Bob cannot communicate after receiving the questions x and y.
- They win if $V(a, b, x, y) = 1$ for some predefined verification function V.

Alice and Bob can agree on a strategy before the game starts, aiming to maximize their winning probability. For a specific strategy S, we denote the winning probability by $\omega(\mathcal{G}, S)$.

Nonlocal Games

A nonlocal game $\mathcal{G}$ is a classical interaction between a referee R and two cooperating players, Alice and Bob:



- Alice and Bob cannot communicate after receiving the questions x and y.
- They win if $V(a, b, x, y) = 1$ for some predefined verification function V.

Alice and Bob can agree on a strategy before the game starts, aiming to maximize their winning probability. For a specific strategy S, we denote the winning probability by $\omega(\mathcal{G}, S)$. The *classical value*

$$\omega_c(\mathcal{G}) := \sup_{S \in \mathcal{S}_c} \omega(\mathcal{G}, S)$$

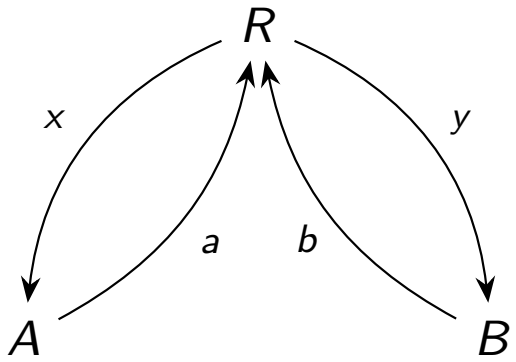
is the maximal winning probability using classical strategies.
The *quantum value*

$$\omega_q(\mathcal{G}) := \sup_{S \in \mathcal{S}_q} \omega(\mathcal{G}, S)$$

is the maximal winning probability using quantum strategies.

Clauser-Horne-Shimony-Holt (CHSH) Game

CHSH Game:

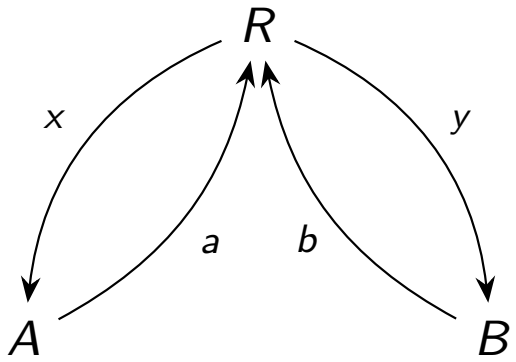


- Questions and answers: $a, b, x, y \in \{0, 1\}$
- Winning condition: $a \oplus b = x \cdot y$

x	y	winning condition
0	0	$a = b$
0	1	$a = b$
1	0	$a = b$
1	1	$a \neq b$

Clauser-Horne-Shimony-Holt (CHSH) Game

CHSH Game:



- Questions and answers: $a, b, x, y \in \{0, 1\}$
- Winning condition: $a \oplus b = x \cdot y$

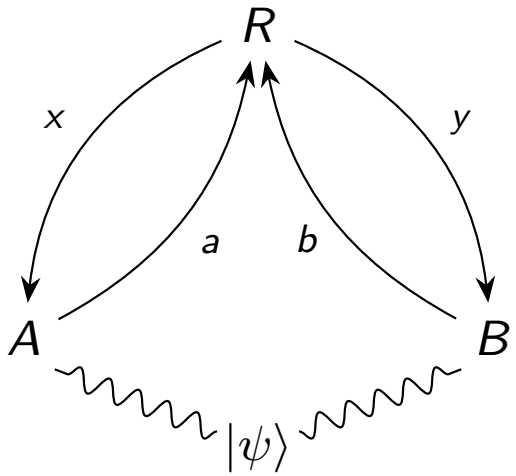
x	y	winning condition
0	0	$a = b$
0	1	$a = b$
1	0	$a = b$
1	1	$a \neq b$

For the classical value, we have

$$\omega_c(\mathcal{G}_{\mathsf{CHSH}}) = 75\%.$$

Clauser-Horne-Shimony-Holt (CHSH) Game

CHSH Game:



- Questions and answers: $a, b, x, y \in \{0, 1\}$
- Winning condition: $a \oplus b = x \cdot y$

x	y	winning condition
0	0	$a = b$
0	1	$a = b$
1	0	$a = b$
1	1	$a \neq b$

For the classical value, we have

$$\omega_c(\mathcal{G}_{\mathsf{CHSH}}) = 75\%.$$

For the quantum value, we have

$$\omega_q(\mathcal{G}_{\mathsf{CHSH}}) = \frac{1}{2} + \frac{1}{2\sqrt{2}} \approx 85\%.$$

Kalai-Lombardi-Vaikuntanathan-Yang (KLVY) Compiler

The KLVY compiler transforms any nonlocal game $\mathcal{G}$ into an interactive protocol $\mathcal{G}_{\text{comp}}$ involving a single computationally bounded player and one referee.

Kalai-Lombardi-Vaikuntanathan-Yang (KLVY) Compiler

The KLVY compiler transforms any nonlocal game $\mathcal{G}$ into an interactive protocol $\mathcal{G}_{\text{comp}}$ involving a single computationally bounded player and one referee.

Motivation for compiling nonlocal games:

- In practice, difficult to enforce that players do not communicate.
- The KLVY compiler provides a modular framework for constructing quantum cryptographic protocols!

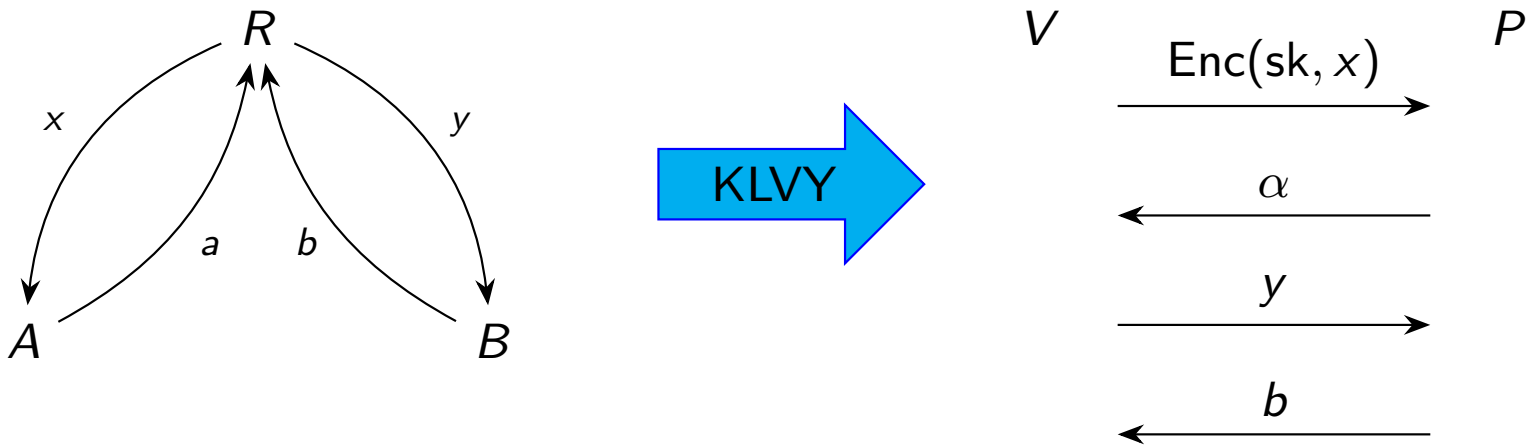
Kalai-Lombardi-Vaikuntanathan-Yang (KLVY) Compiler

The KLVY compiler transforms any nonlocal game $\mathcal{G}$ into an interactive protocol $\mathcal{G}_{\text{comp}}$ involving a single computationally bounded player and one referee.

Motivation for compiling nonlocal games:

- In practice, difficult to enforce that players do not communicate.
- The KLVY compiler provides a modular framework for constructing quantum cryptographic protocols!

The KLVY compiler relies on a *quantum homomorphic encryption* scheme $\text{QHE} = (\text{Gen}, \text{Enc}, \text{Eval}, \text{Dec})$.



The verifier V decrypts α to get $a := \text{Dec}(\text{sk}, \alpha)$, and then checks (a, b, x, y) using the verification function of the underlying nonlocal game.

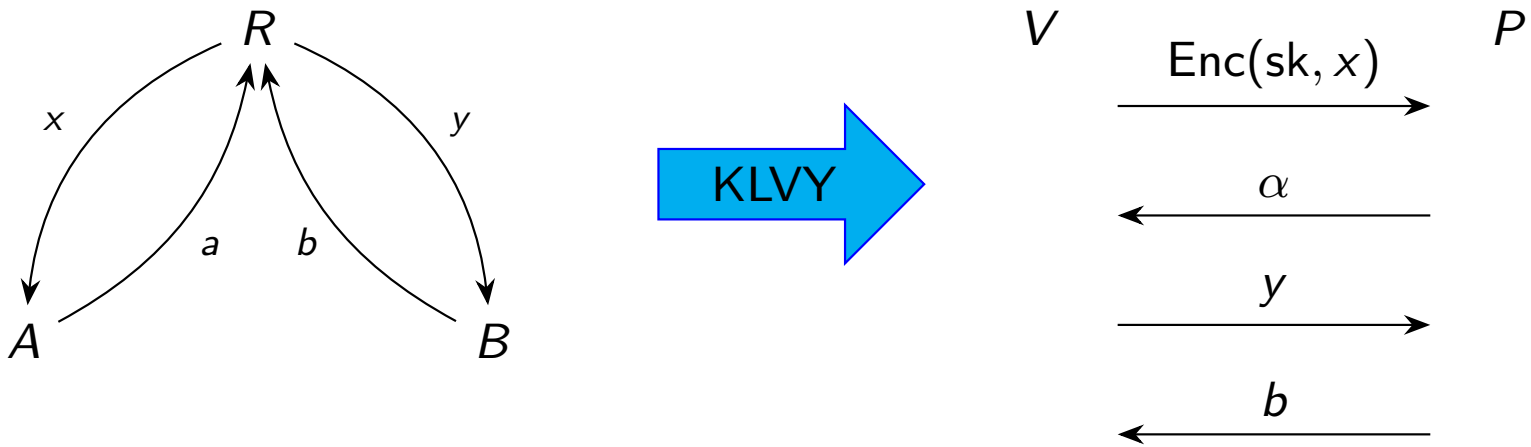
Kalai-Lombardi-Vaikuntanathan-Yang (KLVY) Compiler

The KLVY compiler transforms any nonlocal game $\mathcal{G}$ into an interactive protocol $\mathcal{G}_{\text{comp}}$ involving a single computationally bounded player and one referee.

Motivation for compiling nonlocal games:

- In practice, difficult to enforce that players do not communicate.
- The KLVY compiler provides a modular framework for constructing quantum cryptographic protocols!

The KLVY compiler relies on a *quantum homomorphic encryption* scheme $\text{QHE} = (\text{Gen}, \text{Enc}, \text{Eval}, \text{Dec})$.



The verifier V decrypts α to get $a := \text{Dec}(\text{sk}, \alpha)$, and then checks (a, b, x, y) using the verification function of the underlying nonlocal game.

What can we say about the winning probabilities of $\mathcal{G}_{\text{comp}}$?

Properties of the KLVY Compiler - Part I

Let $\mathcal{G}$ be a nonlocal game, and let $\mathcal{G}_{\text{comp}}$ be the compiled game under the KLVY compiler.

Theorem (Quantum Completeness [KLVY22])

For every quantum strategy S for $\mathcal{G}$, there exists a quantum strategy S_{comp} for $\mathcal{G}_{\text{comp}}$ with

$$\omega_\lambda(\mathcal{G}_{\text{comp}}, S_{\text{comp}}) \geq \omega_q(\mathcal{G}, S) - \text{negl}(\lambda).$$

Achieved by running the strategy S sequentially: first evaluating Alice's circuit on the encrypted question using the homomorphic property of the QHE, and then performing Bob's circuit in the clear on the remaining state.

Properties of the KLVY Compiler - Part I

Let $\mathcal{G}$ be a nonlocal game, and let $\mathcal{G}_{\text{comp}}$ be the compiled game under the KLVY compiler.

Theorem (Quantum Completeness [KLVY22])

For every quantum strategy S for $\mathcal{G}$, there exists a quantum strategy S_{comp} for $\mathcal{G}_{\text{comp}}$ with

$$\omega_\lambda(\mathcal{G}_{\text{comp}}, S_{\text{comp}}) \geq \omega_q(\mathcal{G}, S) - \text{negl}(\lambda).$$

Achieved by running the strategy S sequentially: first evaluating Alice's circuit on the encrypted question using the homomorphic property of the QHE, and then performing Bob's circuit in the clear on the remaining state.

Theorem (Classical Soundness [KLVY22])

For every classical strategy S for $\mathcal{G}_{\text{comp}}$, we have

$$\omega_\lambda(\mathcal{G}_{\text{comp}}, S) \leq \omega_c(\mathcal{G}) + \text{negl}(\lambda).$$

Achieved by considering any single classical prover in the compiled game and constructing two provers for the nonlocal game by rewinding the classical prover. Spatial separation is ensured by the security of the encryption scheme.

Properties of the KLVY Compiler - Part I

Let $\mathcal{G}$ be a nonlocal game, and let $\mathcal{G}_{\text{comp}}$ be the compiled game under the KLVY compiler.

Theorem (Quantum Completeness [KLVY22])

For every quantum strategy S for $\mathcal{G}$, there exists a quantum strategy S_{comp} for $\mathcal{G}_{\text{comp}}$ with

$$\omega_\lambda(\mathcal{G}_{\text{comp}}, S_{\text{comp}}) \geq \omega_q(\mathcal{G}, S) - \text{negl}(\lambda).$$

Achieved by running the strategy S sequentially: first evaluating Alice's circuit on the encrypted question using the homomorphic property of the QHE, and then performing Bob's circuit in the clear on the remaining state.

Theorem (Classical Soundness [KLVY22])

For every classical strategy S for $\mathcal{G}_{\text{comp}}$, we have

$$\omega_\lambda(\mathcal{G}_{\text{comp}}, S) \leq \omega_c(\mathcal{G}) + \text{negl}(\lambda).$$

Achieved by considering any single classical prover in the compiled game and constructing two provers for the nonlocal game by rewinding the classical prover. Spatial separation is ensured by the security of the encryption scheme.

These two results enable a variety of new protocols for *Proofs of Quantumness*.

Properties of the KLVY Compiler - Part II

What about *quantum soundness*?

The work by Natarajan and Zhang [NZ23] established a quantum soundness result for the compiled CHSH game, enabling them to produce a new protocol for *Classical Verification of Quantum Computation* using a QFHE scheme as a black box!

What about *quantum soundness*?

The work by Natarajan and Zhang [NZ23] established a quantum soundness result for the compiled CHSH game, enabling them to produce a new protocol for *Classical Verification of Quantum Computation* using a QFHE scheme as a black box!

Subsequent works established a bound on the quantum value of all compiled nonlocal games:

Theorem (Quantum Soundness [KMPSW24])

Let $\mathcal{G}$ be any two-player nonlocal game, and let $\mathcal{G}_{\text{comp}}$ be the compiled game under the KLVY compiler.
The winning probability of any quantum prover cannot exceed the quantum commuting operator value of $\mathcal{G}$ by more than any constant for a sufficiently large security parameter.
More precisely:
Let S be any quantum strategy for $\mathcal{G}_{\text{comp}}$. Then it holds that

$$\limsup_{\lambda \to \infty} \omega_\lambda(\mathcal{G}_{\text{comp}}, S) \leq \omega_{qc}(\mathcal{G}).$$

A New Compiler for Nonlocal Games

Why consider constructing another compiler?

A New Compiler for Nonlocal Games

Why consider constructing another compiler?

- KLVY compiler relies on the existence of a QFHE scheme, which, until recently, was only constructed from the LWE assumption. Would be nice to have compilers with similar properties, but constructed from different (potentially weaker) cryptographic assumptions.

A New Compiler for Nonlocal Games

Why consider constructing another compiler?

- KLVY compiler relies on the existence of a QFHE scheme, which, until recently, was only constructed from the LWE assumption. Would be nice to have compilers with similar properties, but constructed from different (potentially weaker) cryptographic assumptions.
- To find more *noisy intermediate-scale quantum* (NISQ) era friendly compilers.

A New Compiler for Nonlocal Games

Why consider constructing another compiler?

- KLVY compiler relies on the existence of a QFHE scheme, which, until recently, was only constructed from the LWE assumption. Would be nice to have compilers with similar properties, but constructed from different (potentially weaker) cryptographic assumptions.
- To find more *noisy intermediate-scale quantum* (NISQ) era friendly compilers.
- To gain a better theoretical understanding of the compilation process. Evidence that the functionality offered by QFHE is not necessary, prompting the question of whether this structure is essential for compiled nonlocal games.

Our Results

- We present a novel compiler built upon the framework of measurement-based quantum computation. The compiler can be instantiated assuming the existence of any plain trapdoor claw-free function, which can be constructed from various computational assumptions.

Our Results

- We present a novel compiler built upon the framework of measurement-based quantum computation. The compiler can be instantiated assuming the existence of any plain trapdoor claw-free function, which can be constructed from various computational assumptions.
- Our construction relies on two cryptographic primitives:
  (1) *Blind Remote State Preparation*:
  We introduce a new blind remote state preparation protocol that is constructed from any plain trapdoor claw-free function.
  (2) *Half-Blind Quantum Computation*:
  We provide a generalization of the universal blind quantum computation protocol by Broadbent, Fitzsimons, and Kashefi [BFK09]. This generalization allows a client to perform computations on arbitrary quantum states held by a server while preserving the blindness of the computation.

Our Results

- We present a novel compiler built upon the framework of measurement-based quantum computation. The compiler can be instantiated assuming the existence of any plain trapdoor claw-free function, which can be constructed from various computational assumptions.
- Our construction relies on two cryptographic primitives:
  (1) *Blind Remote State Preparation*:
  We introduce a new blind remote state preparation protocol that is constructed from any plain trapdoor claw-free function.
  (2) *Half-Blind Quantum Computation*:
  We provide a generalization of the universal blind quantum computation protocol by Broadbent, Fitzsimons, and Kashefi [BFK09]. This generalization allows a client to perform computations on arbitrary quantum states held by a server while preserving the blindness of the computation.
- Our compiler satisfies quantum completeness.
- Our compiler satisfies quantum soundness.

Blind RSP - Definition

A *remote state preparation* (RSP) protocol is a classical interaction between a PPT algorithm V and a QPT algorithm P satisfying *correctness*:

- The protocol successfully terminates with a probability of at least $\frac{1}{\text{poly}(\lambda)}$, where λ is the security parameter. Furthermore, upon successful completion, the honest prover P holds the state

$$Z^b \left|+_\theta\right\rangle := Z^b \frac{1}{\sqrt{2}} \left(\left|0\right\rangle + e^{i\theta} \left|1\right\rangle\right)$$

  for some bit $b \in \{0, 1\}$ and angle $\theta \in \{k \cdot \pi/4 \mid k = 0, \ldots, 7\}$. Meanwhile, the verifier holds the pair (b, θ).

Blind RSP - Definition

A *remote state preparation* (RSP) protocol is a classical interaction between a PPT algorithm V and a QPT algorithm P satisfying *correctness*:

- The protocol successfully terminates with a probability of at least $\frac{1}{\text{poly}(\lambda)}$, where λ is the security parameter. Furthermore, upon successful completion, the honest prover P holds the state

$$Z^b |+_\theta\rangle := Z^b \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i\theta} |1\rangle \right)$$

  for some bit $b \in \{0, 1\}$ and angle $\theta \in \{k \cdot \pi/4 \mid k = 0, \ldots, 7\}$. Meanwhile, the verifier holds the pair (b, θ).

For the security definition, we call such an RSP protocol *blind* if:

- Any malicious QPT prover P^* can guess θ at the end of the protocol only with a probability negligibly close to $\frac{1}{8}$.

Blind RSP - Construction (Part I)

Our blind RSP protocol relies on the existence of a plain trapdoor claw-free function.

- A *Trapdoor Claw-Free Function* (TCF) is a family of injective function pairs, along with a trapdoor $(f_0, f_1, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda)$, sharing the same domain and range, i.e., $(f_0, f_1) : \mathcal{X} \to \mathcal{Y}$.
  Claw-freeness refers to the property that, without the trapdoor, it is infeasible for a QPT algorithm to find a *claw*, i.e., two elements $x_0, x_1 \in \mathcal{X}$ such that $f_0(x_0) = f_1(x_1)$. However, with access to the trapdoor, it becomes possible to efficiently invert an image $y \in \mathcal{Y}$ to obtain a claw (x_0, x_1) such that $f_0(x_0) = f_1(x_1) = y$.

Blind RSP - Construction (Part I)

Our blind RSP protocol relies on the existence of a plain trapdoor claw-free function.

- A *Trapdoor Claw-Free Function* (TCF) is a family of injective function pairs, along with a trapdoor $(f_0, f_1, \mathrm{td}) \leftarrow \mathrm{Gen}(1^\lambda)$, sharing the same domain and range, i.e., $(f_0, f_1) : \mathcal{X} \to \mathcal{Y}$.
  Claw-freeness refers to the property that, without the trapdoor, it is infeasible for a QPT algorithm to find a *claw*, i.e., two elements $x_0, x_1 \in \mathcal{X}$ such that $f_0(x_0) = f_1(x_1)$. However, with access to the trapdoor, it becomes possible to efficiently invert an image $y \in \mathcal{Y}$ to obtain a claw (x_0, x_1) such that $f_0(x_0) = f_1(x_1) = y$.
- There are several constructions of TCFs based on the *Learning with Errors* (LWE) problem [BCMVV18], the *Ring-LWE* assumption [BKVV20], and *general cryptographic group actions*, such as isogenies on elliptic curves [AMR22].

Blind RSP - Construction (Part II)

Subroutine:

- (Input) The subroutine is parameterized by an integer n, and a state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ held by the prover P.
- (Output) At the end of the interaction, the verifier holds $(b, \theta) \in \{0,1\} \times \{0,1,2\}$, and the honest prover holds $\alpha|0\rangle + \beta(-1)^b \omega_n^\theta |1\rangle$.

Blind RSP - Construction (Part II)

Subroutine:

- (Input) The subroutine is parameterized by an integer n, and a state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ held by the prover P.
- (Output) At the end of the interaction, the verifier holds $(b, \theta) \in \{0,1\} \times \{0,1,2\}$, and the honest prover holds $\alpha|0\rangle + \beta(-1)^b \omega_n^\theta |1\rangle$.

Blind RSP Protocol: Use the subroutine three times:

- Subroutine with $n = 2$ and $|+\rangle$.
  Let (b_1, θ_1) be the output of V, and $|\psi_1\rangle$ the output state of P.
- Subroutine with $n = 4$ and $|\psi_1\rangle$.
  Let (b_2, θ_2) be the output of V, and $|\psi_2\rangle$ the output state of P.
- Subroutine with $n = 8$ and $|\psi_2\rangle$.
  Let (b_3, θ_3) be the output of V, and $|\psi_3\rangle$ the output state of P.

The prover P holds the final state

$$|\psi_3\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^b \omega_8^\theta |1\rangle) = Z^b \left|+_{\theta \cdot \pi/4}\right\rangle,$$

The verifier V holds

$$b := b_1 \oplus b_2 \oplus b_3 \text{ and } \theta := 4\theta_1 + 2\theta_2 + \theta_3 \mod 8.$$

Blind RSP - Construction (Part III)

Subroutine parameterized by n and state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ held by the prover:

V		$P(\|\psi\rangle)$
$((f_0, f_1), \text{td}) \leftarrow \text{Gen}(1^\lambda)$	$\xrightarrow{(f_0, f_1)}$	Compute claw state $\alpha\|0, x_0\rangle + \beta\|1, x_1\rangle$ with $f_0(x_0) = f_1(x_1) = y$
$(x_0, x_1) \leftarrow \text{Invert}(\text{td}, y)$	$\xleftarrow{y}$	
$r_0, r_1 \leftarrow\!\!\$\ \{0,1\}^{\text{poly}(\lambda)}$	$\xrightarrow{(r_0, r_1)}$	1. Compute $\alpha\|0, x_0, -x_0 r_0\rangle + \beta\|1, x_1, x_1 r_1\rangle$ 2. Apply QFT_n to last register and measure; abort if outcome $\neq 1$. $\alpha\|0, x_0\rangle + \omega_n^{x_0 r_0 + x_1 r_1}\beta\|1, x_1\rangle$ 3. Measure last register in Hadamard basis with outcome $d \in \{0,1\}^{p(\lambda)}$. $\alpha\|0\rangle + \beta(-1)^{d\cdot(x_0 \oplus x_1)}\omega_n^{x_0 r_0 + x_1 r_1}\|1\rangle$
$b := d \cdot (x_0 \oplus x_1)$ $\theta := x_0 r_0 + x_1 r_1$	$\xleftarrow{d}$	

Blind RSP - Correctness and Blindness

- Correctness of the main protocol follows immediately.
- Blindness is established by expressing θ as a sum of terms in the form

$$x_0 r_0 \oplus x_1 r_1 = (x_0 \parallel x_1) \cdot (r_0 \parallel r_1),$$

which allows for the repeated application of the *quantum Goldreich-Levin theorem* to reduce blindness to the claw-freeness of the TCF.

Recap of MBQC

Before proceeding to the half-blind quantum computation protocol, we recap the *measurement-based quantum computation* (MBQC) model.

Task: Compute the quantum state $U|+\rangle^{\otimes n}$.

Recap of MBQC

Before proceeding to the half-blind quantum computation protocol, we recap the *measurement-based quantum computation* (MBQC) model.

Task: Compute the quantum state $U|+\rangle^{\otimes n}$.

In the quantum circuit model:

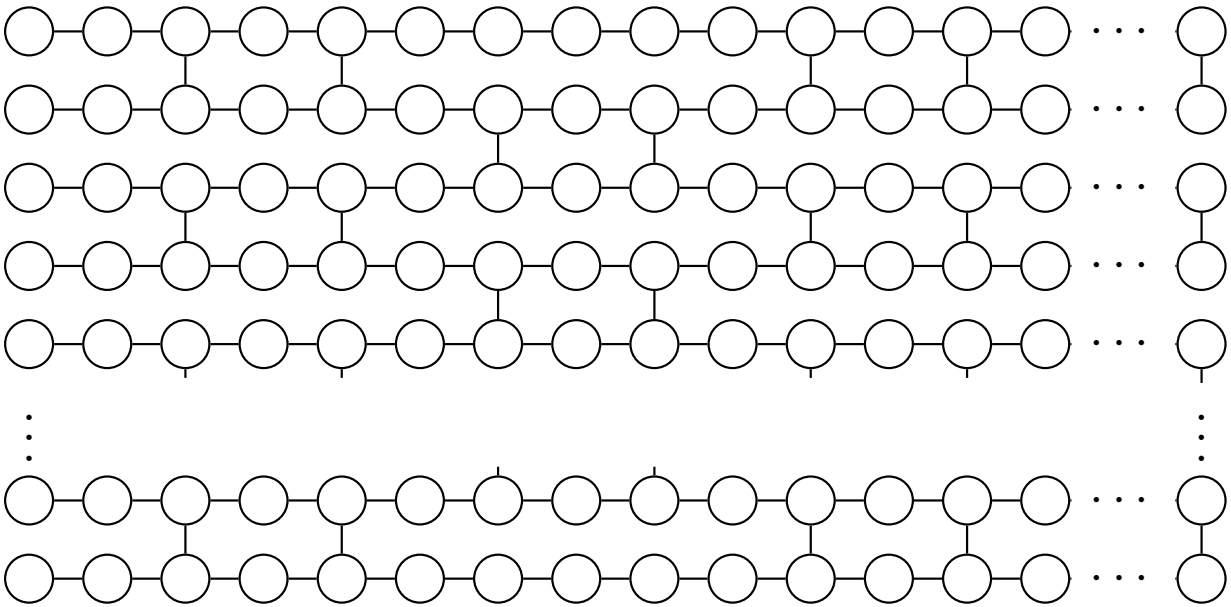
- Efficiently approximate U using a quantum circuit C that operates on n qubits and consists of gates chosen from the universal quantum gate set $\{\mathrm{CX}, H, T\}$.
- Initialize qubits in $|0\rangle^{\otimes n}$, apply $H^{\otimes n}$, and then execute the sequence of gates that approximate U.

Recap of MBQC

Task: Compute the quantum state $U|+\rangle^{\otimes n}$.

In the MBQC model:

- (State Preparation) Prepare multiple $|+\rangle$ states and entangle them in a specific way using CZ to construct the *brickwork state*:

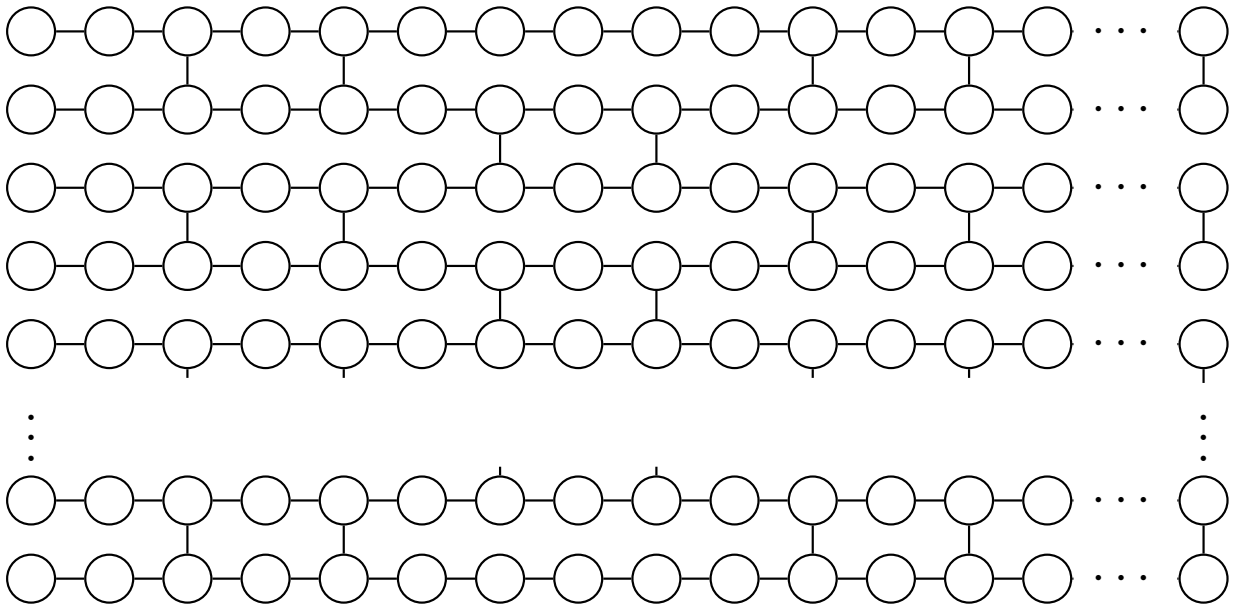


Recap of MBQC

Task: Compute the quantum state $U|+\rangle^{\otimes n}$.

In the MBQC model:

- (State Preparation) Prepare multiple $|+\rangle$ states and entangle them in a specific way using CZ to construct the *brickwork state*:



- (Computation) Single-qubit measurements are performed in a specific basis, starting with the leftmost column and proceeding from top to bottom, then continuing with the subsequent columns until all but the final column have been measured. Finally, the qubits in the last column remain in the desired state (up to local Pauli operators).

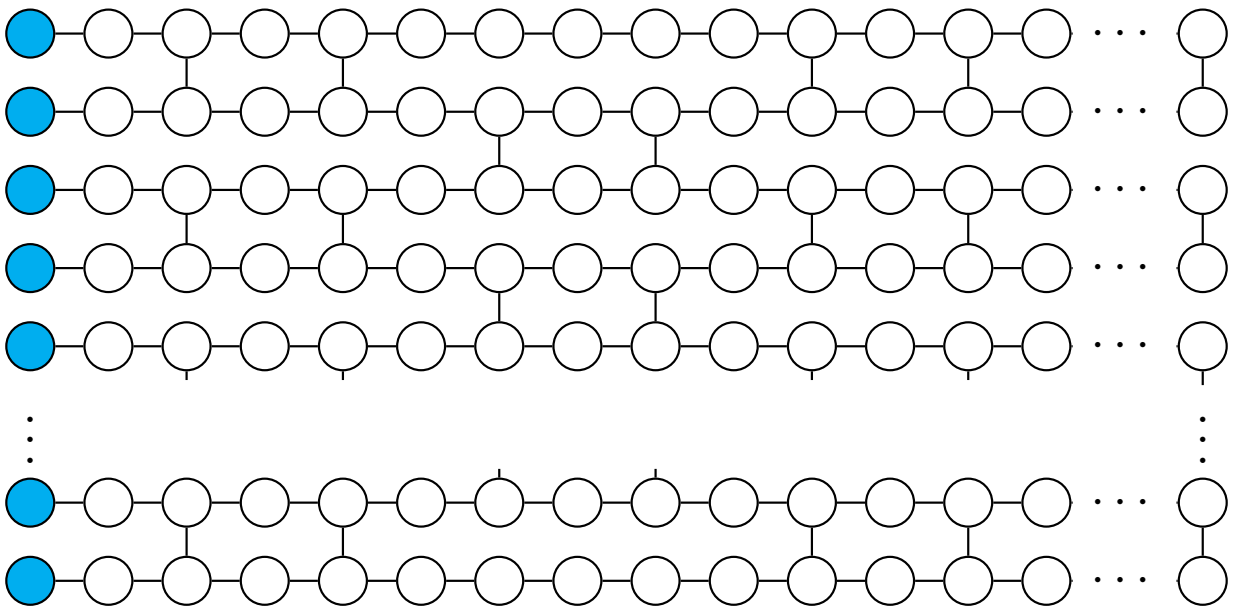
Recap of MBQC

The *universality* of the brickwork state ensures that for each unitary U, such a measurement pattern exists.

Recap of MBQC

The *universality* of the brickwork state ensures that for each unitary U, such a measurement pattern exists.
Let $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n'}$ be an n'-qubit quantum state with $n' \geq n$. By replacing the qubits in the first layer with the first n qubits of $|\psi\rangle$, this procedure implements

$$(U \otimes I)\,|\psi\rangle\,.$$



Recap of UBQC

We recap the *Universal Blind Quantum Computation* (UBQC) protocol by Broadbent, Fitzsimons, and Kashefi [BFK09].

Recap of UBQC

We recap the *Universal Blind Quantum Computation* (UBQC) protocol by Broadbent, Fitzsimons, and Kashefi [BFK09].

- UBQC involves a client and a powerful quantum server. The client requests the server to apply an $n \times n$ unitary U to $|+\rangle^{\otimes n}$, resulting in $U|+\rangle^{\otimes n}$, while keeping U unknown to the server.

Recap of UBQC

We recap the *Universal Blind Quantum Computation* (UBQC) protocol by Broadbent, Fitzsimons, and Kashefi [BFK09].

- UBQC involves a client and a powerful quantum server. The client requests the server to apply an $n \times n$ unitary U to $|+\rangle^{\otimes n}$, resulting in $U|+\rangle^{\otimes n}$, while keeping U unknown to the server.
  - ◇ Step 1: The client prepares single-qubit states $|+_\theta\rangle := \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}|1\rangle)$ with $\theta \leftarrow\!\!\$\ \{k \cdot \pi/4 \mid k = 0, \ldots, 7\}$ and sends them to the server, keeping θ secret.

Recap of UBQC

We recap the *Universal Blind Quantum Computation* (UBQC) protocol by Broadbent, Fitzsimons, and Kashefi [BFK09].

- UBQC involves a client and a powerful quantum server. The client requests the server to apply an $n \times n$ unitary U to $|+\rangle^{\otimes n}$, resulting in $U|+\rangle^{\otimes n}$, while keeping U unknown to the server.
  - ◇ Step 1: The client prepares single-qubit states $|+_\theta\rangle := \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}|1\rangle)$ with $\theta \leftarrow\!\!\$\ \{k \cdot \pi/4 \mid k = 0, \ldots, 7\}$ and sends them to the server, keeping θ secret.
  - ◇ Step 2: The server prepares n qubits in the $|+\rangle$ state for the last layer and entangles all qubits using CZ according to the brickwork state.

Recap of UBQC

We recap the *Universal Blind Quantum Computation* (UBQC) protocol by Broadbent, Fitzsimons, and Kashefi [BFK09].

- UBQC involves a client and a powerful quantum server. The client requests the server to apply an $n \times n$ unitary U to $|+\rangle^{\otimes n}$, resulting in $U|+\rangle^{\otimes n}$, while keeping U unknown to the server.
  - ◇ Step 1: The client prepares single-qubit states $|+_\theta\rangle := \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}|1\rangle)$ with $\theta \leftarrow\!\!\$\ \{k \cdot \pi/4 \mid k = 0, \ldots, 7\}$ and sends them to the server, keeping θ secret.
  - ◇ Step 2: The server prepares n qubits in the $|+\rangle$ state for the last layer and entangles all qubits using CZ according to the brickwork state.
  - ◇ Step 3: Using the measurement pattern implementing U, the client instructs the server to perform specific measurements. The server reports outcomes and ends with $U|+\rangle^{\otimes n}$, up to local Pauli operators.

Recap of UBQC

We recap the *Universal Blind Quantum Computation* (UBQC) protocol by Broadbent, Fitzsimons, and Kashefi [BFK09].

- UBQC involves a client and a powerful quantum server. The client requests the server to apply an $n \times n$ unitary U to $|+\rangle^{\otimes n}$, resulting in $U|+\rangle^{\otimes n}$, while keeping U unknown to the server.
  - ◇ Step 1: The client prepares single-qubit states $|+_\theta\rangle := \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}|1\rangle)$ with $\theta \leftarrow\$ \{k \cdot \pi/4 \mid k = 0, \ldots, 7\}$ and sends them to the server, keeping θ secret.
  - ◇ Step 2: The server prepares n qubits in the $|+\rangle$ state for the last layer and entangles all qubits using CZ according to the brickwork state.
  - ◇ Step 3: Using the measurement pattern implementing U, the client instructs the server to perform specific measurements. The server reports outcomes and ends with $U|+\rangle^{\otimes n}$, up to local Pauli operators.
- In step 3, the client adjusts the measurement angles to cancel out the randomness introduced in step 1.

Recap of UBQC

We recap the *Universal Blind Quantum Computation* (UBQC) protocol by Broadbent, Fitzsimons, and Kashefi [BFK09].

- UBQC involves a client and a powerful quantum server. The client requests the server to apply an $n \times n$ unitary U to $|+\rangle^{\otimes n}$, resulting in $U|+\rangle^{\otimes n}$, while keeping U unknown to the server.
  - ◇ Step 1: The client prepares single-qubit states $|+_\theta\rangle := \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}|1\rangle)$ with $\theta \leftarrow\$ \{k \cdot \pi/4 \mid k = 0, \ldots, 7\}$ and sends them to the server, keeping θ secret.
  - ◇ Step 2: The server prepares n qubits in the $|+\rangle$ state for the last layer and entangles all qubits using CZ according to the brickwork state.
  - ◇ Step 3: Using the measurement pattern implementing U, the client instructs the server to perform specific measurements. The server reports outcomes and ends with $U|+\rangle^{\otimes n}$, up to local Pauli operators.
- In step 3, the client adjusts the measurement angles to cancel out the randomness introduced in step 1.
- The injected randomness ensures *information-theoretic blindness*: from the server's perspective, the measurement pattern for U appears uniformly random.

HBQC

We generalize the UBQC protocol to apply a unitary U blindly to any quantum state $|\psi\rangle$ held by the server, implementing $(U \otimes I)|\psi\rangle$. This is referred to as *half-blind quantum computation* (HBQC). The HBQC protocol is nearly identical to UBQC, with key modifications:

HBQC

We generalize the UBQC protocol to apply a unitary U blindly to any quantum state $|\psi\rangle$ held by the server, implementing $(U \otimes I)|\psi\rangle$. This is referred to as *half-blind quantum computation* (HBQC). The HBQC protocol is nearly identical to UBQC, with key modifications:

- Instead of using the $|+\rangle^{\otimes n}$ state as the first layer, use the first n qubits of $|\psi\rangle$, resulting in the computation $(U \otimes I)|\psi\rangle$. However, this prevents the client to inject randomness into these qubits, posing a challenge for blindness.

HBQC

We generalize the UBQC protocol to apply a unitary U blindly to any quantum state $|\psi\rangle$ held by the server, implementing $(U \otimes I)|\psi\rangle$. This is referred to as *half-blind quantum computation* (HBQC). The HBQC protocol is nearly identical to UBQC, with key modifications:

- Instead of using the $|+\rangle^{\otimes n}$ state as the first layer, use the first n qubits of $|\psi\rangle$, resulting in the computation $(U \otimes I)|\psi\rangle$. However, this prevents the client to inject randomness into these qubits, posing a challenge for blindness.
- To circumvent this, eight additional layers, prepared by the server, are introduced at the beginning.

HBQC

We generalize the UBQC protocol to apply a unitary U blindly to any quantum state $|\psi\rangle$ held by the server, implementing $(U \otimes I)|\psi\rangle$. This is referred to as *half-blind quantum computation* (HBQC). The HBQC protocol is nearly identical to UBQC, with key modifications:

- Instead of using the $|+\rangle^{\otimes n}$ state as the first layer, use the first n qubits of $|\psi\rangle$, resulting in the computation $(U \otimes I)|\psi\rangle$. However, this prevents the client to inject randomness into these qubits, posing a challenge for blindness.
- To circumvent this, eight additional layers, prepared by the server, are introduced at the beginning.
- These first eight layers are measured in such a way to implement the identity gate. The remaining qubits follow the original measurement pattern. Note that after the first eight layers, the server measures client-prepared qubits containing injected randomness.

HBQC

We generalize the UBQC protocol to apply a unitary U blindly to any quantum state $|\psi\rangle$ held by the server, implementing $(U \otimes I)|\psi\rangle$. This is referred to as *half-blind quantum computation* (HBQC). The HBQC protocol is nearly identical to UBQC, with key modifications:

- Instead of using the $|+\rangle^{\otimes n}$ state as the first layer, use the first n qubits of $|\psi\rangle$, resulting in the computation $(U \otimes I)|\psi\rangle$. However, this prevents the client to inject randomness into these qubits, posing a challenge for blindness.
- To circumvent this, eight additional layers, prepared by the server, are introduced at the beginning.
- These first eight layers are measured in such a way to implement the identity gate. The remaining qubits follow the original measurement pattern. Note that after the first eight layers, the server measures client-prepared qubits containing injected randomness.

Correctness and information-theoretic blindness are ensured as in UBQC. The intuition for blindness is that the protocol effectively teleports $|\psi\rangle$ to a position where randomness has already been injected, replicating the effect of directly injecting randomness into $|\psi\rangle$.

Description of the New Compiler

Replace the quantum communication in HBQC with the blind RSP protocol to achieve purely classical interaction. We call this the *classical half-blind quantum computation* (CHBQC) protocol. Information-theoretical blindness is replaced by computational blindness.

Description of the New Compiler

Replace the quantum communication in HBQC with the blind RSP protocol to achieve purely classical interaction. We call this the *classical half-blind quantum computation (CHBQC)* protocol. Information-theoretical blindness is replaced by computational blindness.

Our compiler: Let $\{U_x\}_x$ represent the unitaries corresponding to Alice's strategy for the game $\mathcal{G}$. The prover and verifier execute the following interactive protocol:

- ◇ The verifier samples a question pair (x, y).
- ◇ The verifier and prover engage in the CHBQC protocol. The verifier's input is U_x, while the prover's state $|\psi\rangle$ is arbitrary. Let a' denote the prover's output, and let a be the verifier's output derived from a'.
- ◇ The verifier sends y to the prover in plaintext.
- ◇ The prover responds with some b.
- ◇ The verifier accepts if $V(a, b, x, y) = 1$.

Description of the New Compiler

Replace the quantum communication in HBQC with the blind RSP protocol to achieve purely classical interaction. We call this the *classical half-blind quantum computation (CHBQC)* protocol. Information-theoretical blindness is replaced by computational blindness.

Our compiler: Let $\{U_x\}_x$ represent the unitaries corresponding to Alice's strategy for the game $\mathcal{G}$. The prover and verifier execute the following interactive protocol:

- ◇ The verifier samples a question pair (x, y).
- ◇ The verifier and prover engage in the CHBQC protocol. The verifier's input is U_x, while the prover's state $|\psi\rangle$ is arbitrary. Let a' denote the prover's output, and let a be the verifier's output derived from a'.
- ◇ The verifier sends y to the prover in plaintext.
- ◇ The prover responds with some b.
- ◇ The verifier accepts if $V(a, b, x, y) = 1$.

What can we say about the winning probabilities of $\mathcal{G}_{\text{comp}}$?

Properties of the New Compiler

Let $\mathcal{G}$ be any two-player nonlocal game, and let $\mathcal{G}_{\text{comp}}$ be the corresponding compiled game under our compiler

Theorem (Quantum Completeness)

For every quantum strategy S for $\mathcal{G}$, there exists a quantum strategy S_{comp} for $\mathcal{G}_{\text{comp}}$ with

$$\omega_\lambda(\mathcal{G}_{\text{comp}}, S_{\text{comp}}) \geq \omega_q(\mathcal{G}, S) - \text{negl}(\lambda).$$

Achieved by running the strategy S sequentially: first, blindly applying Alice's unitary, measuring her qubits, then applying Bob's unitary, and measuring his qubits.

Properties of the New Compiler

Let $\mathcal{G}$ be any two-player nonlocal game, and let $\mathcal{G}_{\text{comp}}$ be the corresponding compiled game under our compiler

Theorem (Quantum Completeness)

For every quantum strategy S for $\mathcal{G}$, there exists a quantum strategy S_{comp} for $\mathcal{G}_{\text{comp}}$ with

$$\omega_\lambda(\mathcal{G}_{\text{comp}}, S_{\text{comp}}) \geq \omega_q(\mathcal{G}, S) - \text{negl}(\lambda).$$

Achieved by running the strategy S sequentially: first, blindly applying Alice's unitary, measuring her qubits, then applying Bob's unitary, and measuring his qubits.

Theorem (Quantum Soundness)

Let S be any quantum strategy for $\mathcal{G}_{\text{comp}}$. Then it holds that

$$\limsup_{\lambda \to \infty} \omega_\lambda(\mathcal{G}_{\text{comp}}, S) \leq \omega_{qc}(\mathcal{G}).$$

To prove this, we primarily leverage results from previous works in [NZ23, KMPSW24]. Only need to reprove the theorems where the security properties of the QHE scheme are used, replacing them with the computational blindness properties of this compiler.

Summary

- Constructed a new compiler:
  Transforms any nonlocal game into a single-prover interactive protocol by combining our blind RSP protocol with our generalized UBQC protocol.
  Relies solely on the existence of plain quantum-secure TCFs.
- Compiler satisfies quantum completeness.
- Compiler satisfies quantum soundness.

Thank you for your attention!
Questions?