Compiled Nonlocal Games from Any TCF

K. Bacho[1]   A. Kulpe[2]   G. Malavolta[3]   S. Schmidt[2]   M. Walter[2]

[1]University of Edinburgh

[2]Ruhr-University Bochum

[3]Bocconi University

QUASAR Seminar, 2025-05-16

Table of Contents

Motivation: Quantum Advantage

$\boxed{\text{Quantum Computer?}}$

Motivation: Quantum Advantage

Quantum Computer?

- How to test that this box is a quantum computer?

Motivation: Quantum Advantage

Quantum Computer?

- How to test that this box is a quantum computer?
  - Ask it to *factor* an RSA-2048 number

Motivation: Quantum Advantage

Quantum Computer?

- How to test that this box is a quantum computer?
  - Ask it to *factor* an RSA-2048 number
    - We would be impressed
    - Maybe factoring is in P?

Motivation: Quantum Advantage

Quantum Computer?

- How to test that this box is a quantum computer?
  - Ask it to *factor* an RSA-2048 number
  - Run some quantum protocol (i.e. *QKD*) between two boxes

Motivation: Quantum Advantage

Quantum Computer?

- How to test that this box is a quantum computer?
  - Ask it to *factor* an RSA-2048 number
  - Run some quantum protocol (i.e. *QKD*) between two boxes
    - Practical
    - Need *two* quantum devices that communicate

Motivation: Quantum Advantage

Quantum Computer?

- How to test that this box is a quantum computer?
  - Ask it to *factor* an RSA-2048 number
  - Run some quantum protocol (i.e. *QKD*) between two boxes
  - Send some *quantum state* to the box and have it apply some operation

Motivation: Quantum Advantage

Quantum Computer?

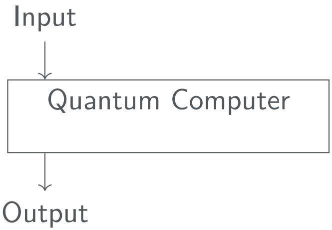
- How to test that this box is a quantum computer?
  - Ask it to *factor* an RSA-2048 number
  - Run some quantum protocol (i.e. *QKD*) between two boxes
  - Send some *quantum state* to the box and have it apply some operation
    - In principle easy
    - Verifier needs to be quantum

Motivation: Quantum Advantage

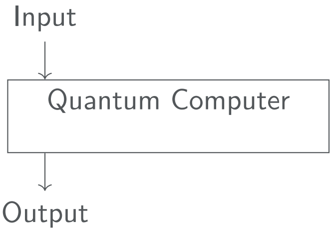
Quantum Computer?

- How to test that this box is a quantum computer?
  - Ask it to *factor* an RSA-2048 number
  - Run some quantum protocol (i.e. *QKD*) between two boxes
  - Send some *quantum state* to the box and have it apply some operation
- Question: Can a *classical* verifier check that the box is quantum?

Motivation: Classically Verifying Quantum Computation

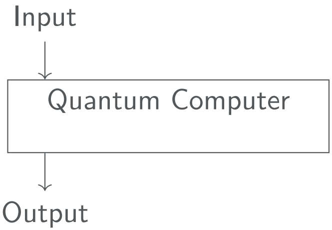


Motivation: Classically Verifying Quantum Computation



- Question: Can a *classical* verifier check that the output is correct, i.e. can we verify the quantum computation *classically*?

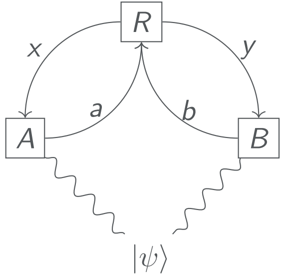
Motivation: Classically Verifying Quantum Computation



- Question: Can a *classical* verifier check that the output is correct, i.e. can we verify the quantum computation *classically*?
- Answer: All this and more is possible with *nonlocal games*!

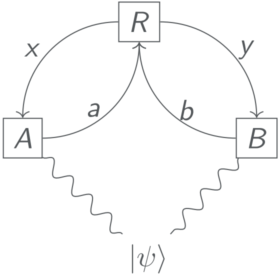
Table of Contents

Nonlocal Games



- Players are not allowed to communicate during the game
- They win if $V(a, b \mid x, y) = 1$

Nonlocal Games



- Players are not allowed to communicate during the game
- They win if $V(a, b \mid x, y) = 1$

Alice and Bob want to maximize their winning probability

$$\omega(S, \mathcal{G}) = \sum_{x,y} \pi(x, y) \sum_{a,b} V(a, b \mid x, y) p(a, b \mid x, y).$$

Strategies

There are different classes of strategies, the optimal winning probability for this class is called a *value*:

	[Bell]	[Ji–Natarajan–Vidick–Wright–Yuen]	[Popescu–Rohrlich]	
ω_c	ω_q	ω_{qc}	ω_{ns}	
Classical Value	Quantum value $H_A \otimes H_B$	Commuting operator value $[A, B] = 0$	Non-signaling value	

They can all be different!

Amazing fact: For certain games, the optimal quantum strategies are "unique" giving a starting point for many applications (self-testing, verifying quantum computation,...)

Nonlocality Approach

Advantages of Nonlocality Approach

- Can be used to verify arbitrary quantum computation [Reichardt-Unger, Vazirani, Grilo]
- No computational assumptions

Disadvantages of Nonlocality Approach

- Need two quantum devices
- Have to ensure nonlocality (the players are not communicating)

Nonlocality Approach

Advantages of Nonlocality Approach

- Can be used to verify arbitrary quantum computation [Reichardt-Unger, Vazirani, Grilo]
- No computational assumptions

Disadvantages of Nonlocality Approach

- Need two quantum devices
- Have to ensure nonlocality (the players are not communicating)

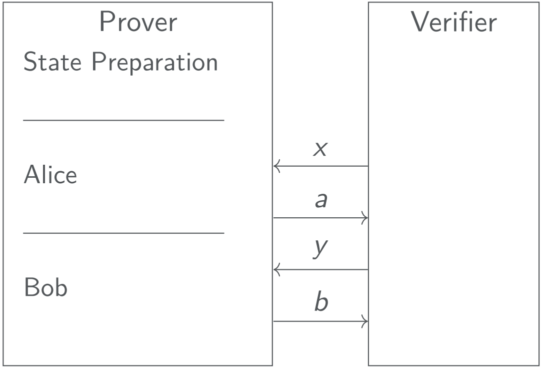
Instead: Can one verify quantum computation by interacting with a single device?

- Yes, under computational assumptions [Mahadev]

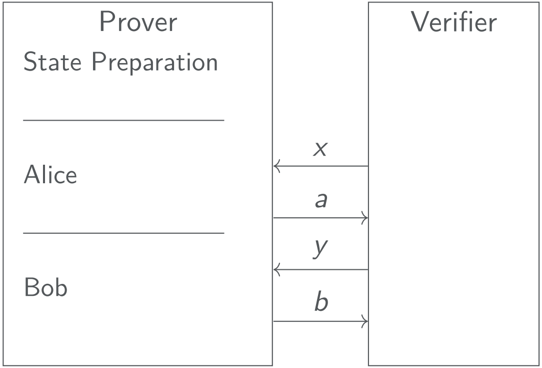
Question

Can we eliminate space-like separation and play a nonlocal game with a single device?

Compiled Nonlocal Game: Naive attempt



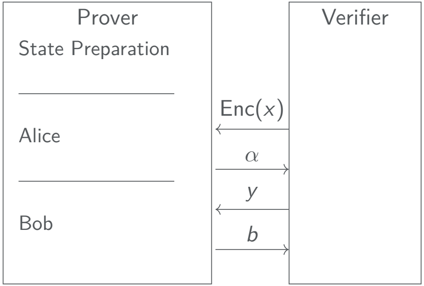
Compiled Nonlocal Game: Naive attempt



Cannot work since it even allows forward signaling!

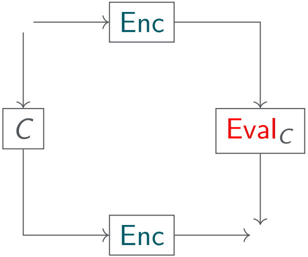
Idea: Use cryptography!

Compiled Nonlocal Game: Clever attempt

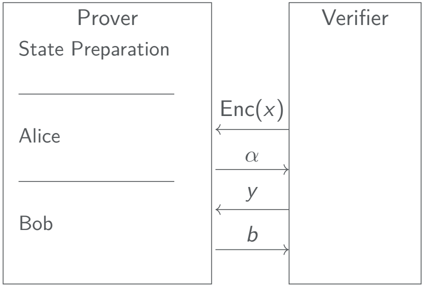


Approach of [Kalai-Lombardi-Vaikuntanathan-Yang]

- Encrypt Alice's question
- Bob's question can be sent in the plain.
- Use quantum homomorphic encryption (QHE) scheme, to allow computing on encrypted data



Compiled Nonlocal Game



- The prover is *computationally bounded*, that is, quantum strategies consist of QPT algorithms
- Using the properties of QHE scheme, the post-measurement states $\rho_x = \sum_a \rho_{xa}$ after Alice's answer are *computationally indistinguishable*, $\rho_{x_1} \approx \rho_{x_2}$ (no QPT algorithm can tell the difference)

Values for Compiled Nonlocal Games

Let $\mathcal{G}$ be a nonlocal game.

Theorem 1 ([KLVY])

1. *Classical soundness: Any classical strategy for the compiled game has winning probability at most* $\omega_c(\mathcal{G}) + \mathsf{negl}(\lambda)$
2. *Quantum completeness: For every quantum strategy* S *of* $\mathcal{G}$ *there exists a quantum strategy for the compiled game with winning probability at least* $\omega(S, \mathcal{G}) - \mathsf{negl}(\lambda)$.

Values for Compiled Nonlocal Games

Let $\mathcal{G}$ be a nonlocal game.

Theorem 1 ([KLVY])

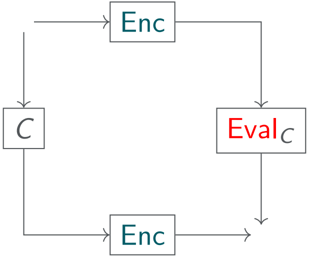
1. *Classical soundness: Any classical strategy for the compiled game has winning probability at most* $\omega_c(\mathcal{G}) + \mathsf{negl}(\lambda)$
2. *Quantum completeness: For every quantum strategy* S *of* $\mathcal{G}$ *there exists a quantum strategy for the compiled game with winning probability at least* $\omega(S, \mathcal{G}) - \mathsf{negl}(\lambda)$.

Theorem 2 ([K-Malavolta-Paddock-Schmidt-Walter])

Quantum Soundness: For large enough security parameter λ*, no QPT strategy can win the compiled nonlocal game with probability exceeding the quantum commuting operator value of the game by any constant:*

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

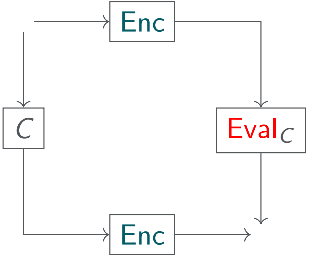
Q(F)HE



Additional properties:

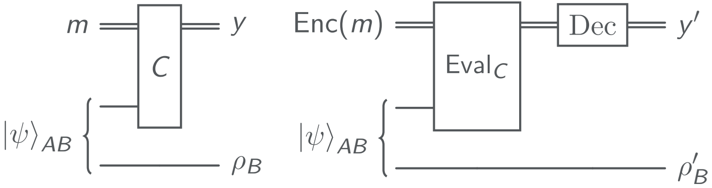
- Evaluation of Alice's circuit from some optimal strategy has to be supported

Q(F)HE



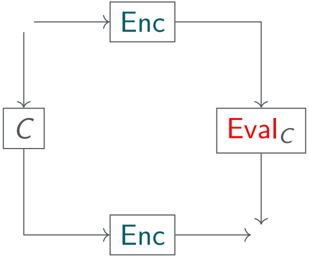
Additional properties:

- Evaluation of Alice's circuit from some optimal strategy has to be supported
- Correctness with auxiliary input



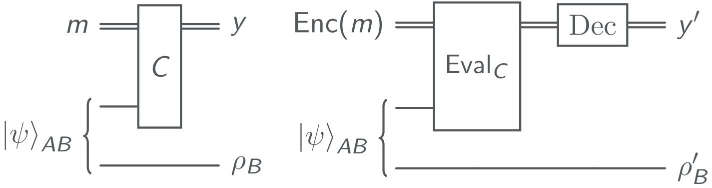
$$(y, \rho_B) \approx_c (y', \rho'_B)$$

Q(F)HE



Additional properties:

- Evaluation of Alice's circuit from some optimal strategy has to be supported
- Correctness with auxiliary input

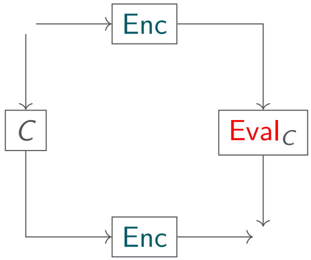


$$(y, \rho_B) \approx_c (y', \rho'_B)$$

- Security

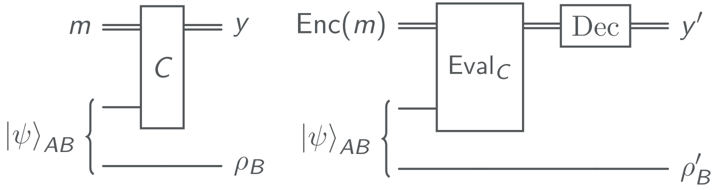
$$\mathsf{Enc}(m) \approx_c \mathsf{Enc}(m') \quad \forall m, m'$$

Q(F)HE



Additional properties:

- Evaluation of Alice's circuit from some optimal strategy has to be supported
- Correctness with auxiliary input



$$(y, \rho_B) \approx_c (y', \rho'_B)$$

- Security

$$\mathrm{Enc}(m) \approx_c \mathrm{Enc}(m') \quad \forall m, m'$$

Question: How to construct such QHE schemes?

Constructions for QHE scheme

Constructions based on

- LWE (special TCF with additional properties) [Mahadev, Brakerski]
- iO + dual-mode TCF [Gupte-Vaikuntanathan]

Constructions for QHE scheme

Constructions based on

- LWE (special TCF with additional properties) [Mahadev, Brakerski]
- iO + dual-mode TCF [Gupte-Vaikuntanathan]

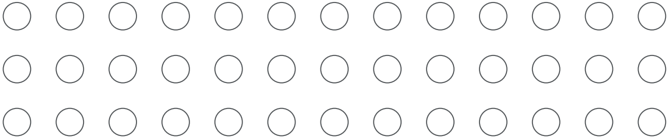
Question: Do we need QHE?
Note that QHE is minimally interactive version of blind classical delegation of quantum computation
Idea: Blindly delegate Alice's computation instead of using QHE

Table of Contents

Universal Blind Quantum Computation [Broadbent-Fitz-Kashefi]

Goal: Blind Delegation of $|+\rangle^{\otimes n} \mapsto U|+\rangle^{\otimes n}$

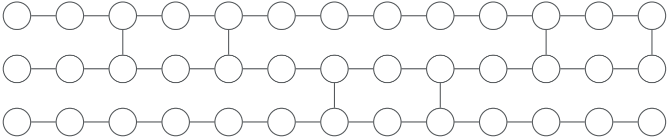
1. Client prepares specific single-qubit states and sends them to the server



Universal Blind Quantum Computation [Broadbent-Fitz-Kashefi]

Goal: Blind Delegation of $|+\rangle^{\otimes n} \mapsto U|+\rangle^{\otimes n}$

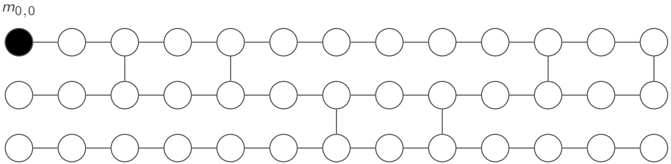
1. Client prepares specific single-qubit states and sends them to the server
2. Server entangles them $\rightarrow$ *brickwork state*



Universal Blind Quantum Computation [Broadbent-Fitz-Kashefi]

Goal: Blind Delegation of $|+\rangle^{\otimes n} \mapsto U|+\rangle^{\otimes n}$

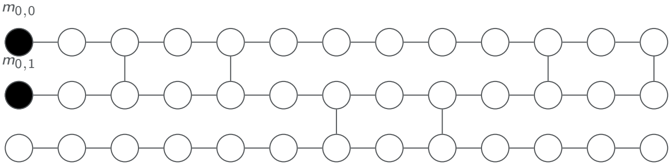
1. Client prepares specific single-qubit states and sends them to the server
2. Server entangles them $\rightarrow$ *brickwork state*
3. Client adaptively chooses measurement basis and server measures and responds with the measurement result



Universal Blind Quantum Computation [Broadbent-Fitz-Kashefi]

Goal: Blind Delegation of $|+\rangle^{\otimes n} \mapsto U|+\rangle^{\otimes n}$

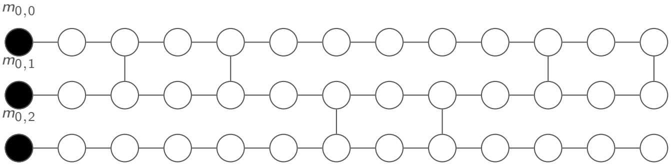
1. Client prepares specific single-qubit states and sends them to the server
2. Server entangles them $\rightarrow$ *brickwork state*
3. Client adaptively chooses measurement basis and server measures and responds with the measurement result



Universal Blind Quantum Computation [Broadbent-Fitz-Kashefi]

Goal: Blind Delegation of $|+\rangle^{\otimes n} \mapsto U|+\rangle^{\otimes n}$

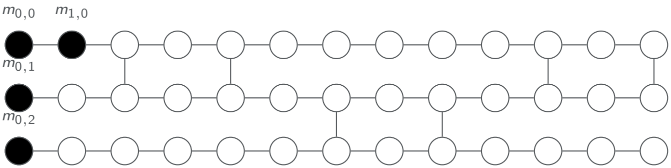
1. Client prepares specific single-qubit states and sends them to the server
2. Server entangles them $\rightarrow$ *brickwork state*
3. Client adaptively chooses measurement basis and server measures and responds with the measurement result



Universal Blind Quantum Computation [Broadbent-Fitz-Kashefi]

Goal: Blind Delegation of $|+\rangle^{\otimes n} \mapsto U|+\rangle^{\otimes n}$

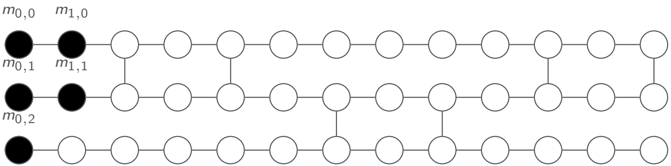
1. Client prepares specific single-qubit states and sends them to the server
2. Server entangles them $\rightarrow$ *brickwork state*
3. Client adaptively chooses measurement basis and server measures and responds with the measurement result



Universal Blind Quantum Computation [Broadbent-Fitz-Kashefi]

Goal: Blind Delegation of $|+\rangle^{\otimes n} \mapsto U|+\rangle^{\otimes n}$

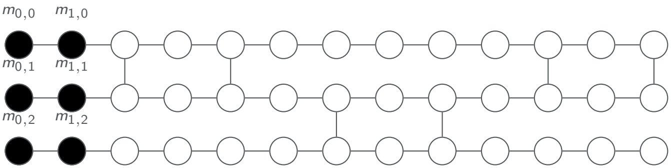
1. Client prepares specific single-qubit states and sends them to the server
2. Server entangles them $\rightarrow$ *brickwork state*
3. Client adaptively chooses measurement basis and server measures and responds with the measurement result



Universal Blind Quantum Computation [Broadbent-Fitz-Kashefi]

Goal: Blind Delegation of $|+\rangle^{\otimes n} \mapsto U|+\rangle^{\otimes n}$

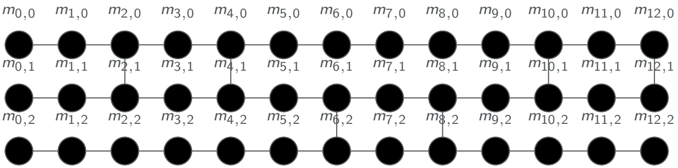
1. Client prepares specific single-qubit states and sends them to the server
2. Server entangles them $\rightarrow$ *brickwork state*
3. Client adaptively chooses measurement basis and server measures and responds with the measurement result



Universal Blind Quantum Computation [Broadbent-Fitz-Kashefi]

Goal: Blind Delegation of $|+\rangle^{\otimes n} \mapsto U|+\rangle^{\otimes n}$

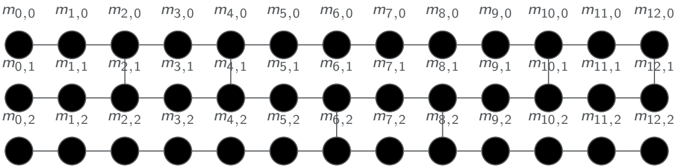
1. Client prepares specific single-qubit states and sends them to the server
2. Server entangles them $\rightarrow$ *brickwork state*
3. Client adaptively chooses measurement basis and server measures and responds with the measurement result



Universal Blind Quantum Computation [Broadbent-Fitz-Kashefi]

Goal: Blind Delegation of $|+\rangle^{\otimes n} \mapsto U|+\rangle^{\otimes n}$

1. Client prepares specific single-qubit states and sends them to the server
2. Server entangles them $\rightarrow$ *brickwork state*
3. Client adaptively chooses measurement basis and server measures and responds with the measurement result



In the end, prover measures his one-time padded state $(X^{x_1} Z^{z_1} \otimes \cdots \otimes X^{x_n} Z^{z_n}) U |+\rangle^{\otimes n}$

Client knows measurement result of $U|+\rangle^{\otimes n}$

Question: Can we use UBQC instead of QHE?

Half-Blind Quantum Computation

- Allow arbitrary input state

$$U\left|\psi\right\rangle \text{ instead of } U\left|+\right\rangle^{\otimes n}$$

$\Rightarrow$ Teleport $\left|\psi\right\rangle$ into brickwork state

Half-Blind Quantum Computation

- Allow arbitrary input state

$$U|\psi\rangle \text{ instead of } U|+\rangle^{\otimes n}$$

$\Rightarrow$ Teleport $|\psi\rangle$ into brickwork state

- Preserve entanglement between Alice and Bob

$$(U_A \otimes I)|\psi\rangle_{AB}$$

$\Rightarrow$ Pauli errors do not propagate to Bob's subsystem

Half-Blind Quantum Computation

- Allow arbitrary input state

$$U|\psi\rangle \text{ instead of } U|+\rangle^{\otimes n}$$

$\Rightarrow$ Teleport $|\psi\rangle$ into brickwork state

- Preserve entanglement between Alice and Bob

$$(U_A \otimes I)|\psi\rangle_{AB}$$

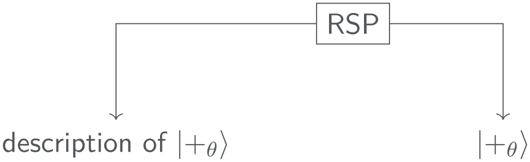
$\Rightarrow$ Pauli errors do not propagate to Bob's subsystem

- Quantum Communication

$$|+_\theta\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}|1\rangle), \quad \theta \in \left\{0, \frac{\pi}{4}, \dots, \frac{7\pi}{4}\right\}$$

$\Rightarrow$ Let the server blindly prepare these states using only classical communication

Blind Remote State Preparation



RSP is called *blind* if the server does not learn anything about $|+_\theta\rangle$ during the interaction

Blind Remote State Preparation

RSP

description of $|+_\theta\rangle$ $\qquad$ $|+_\theta\rangle$

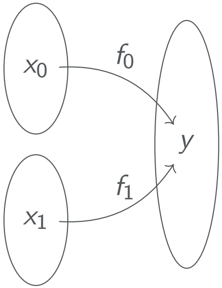
RSP is called *blind* if the server does not learn anything about $|+_\theta\rangle$ during the interaction

Constructions of blind RSP for $|+_\theta\rangle$ based on

- LWE [Gheorghiu-Vidick]

Question: Can we weaken that assumption?
Now: Blind RSP from any TCF

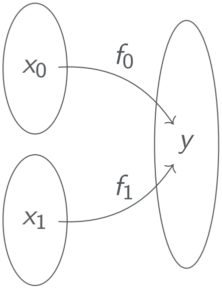
Trapdoor Claw-Free Functions



Trapdoor Claw-Free Function (TCF) pair f_0, f_1

- $((f_0, f_1), \mathrm{td}) \leftarrow \mathrm{Gen}(1^\lambda)$
- f_0, f_1 injective with same image
- Efficient inversion given trapdoor: $(x_0, x_1) \leftarrow \mathrm{Invert}(\mathrm{td}, y)$ s.th. $f_0(x_0) = f_1(x_1) = y$
- Hard to find claw (x_0, x_1) such that $f_0(x_0) = f_1(x_1)$

Trapdoor Claw-Free Functions



Trapdoor Claw-Free Function (TCF) pair f_0, f_1

- $((f_0, f_1), \mathrm{td}) \leftarrow \mathrm{Gen}(1^\lambda)$
- f_0, f_1 injective with same image
- Efficient inversion given trapdoor: $(x_0, x_1) \leftarrow \mathrm{Invert}(\mathrm{td}, y)$ s.th. $f_0(x_0) = f_1(x_1) = y$
- Hard to find claw (x_0, x_1) such that $f_0(x_0) = f_1(x_1)$

Known TCFs:

- LWE
- cryptographic group actions [Alamati-Malavolta-Rahimi]
- ...

Blind RSP from TCFs: Construction I - Subprotocol(1^λ, n, $|\psi\rangle$)

Client		**Server**
$((f_0, f_1), \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda)$	$\xrightarrow{(f_0, f_1)}$	initial state: $\|\psi\rangle = \alpha\|0\rangle + \beta\|1\rangle$

Blind RSP from TCFs: Construction I - Subprotocol$(1^\lambda, n, |\psi\rangle)$

Client		**Server**
$((f_0, f_1), \mathrm{td}) \leftarrow \mathrm{Gen}(1^\lambda)$	$\xrightarrow{(f_0, f_1)}$	initial state: $\|\psi\rangle = \alpha\|0\rangle + \beta\|1\rangle$
	$\xleftarrow{y}$	Prepare $\alpha\|0, x_0\rangle + \beta\|1, x_1\rangle$
		where $f_0(x_0) = f_1(x_1) = y$

Blind RSP from TCFs: Construction I - Subprotocol$(1^\lambda, n, |\psi\rangle)$

Client		**Server**
$((f_0, f_1), \mathrm{td}) \leftarrow \mathrm{Gen}(1^\lambda)$	$\xrightarrow{(f_0, f_1)}$	initial state: $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$
	$\xleftarrow{y}$	Prepare $\alpha\,|0, x_0\rangle + \beta\,|1, x_1\rangle$
		where $f_0(x_0) = f_1(x_1) = y$
$r_0, r_1 \leftarrow \{0,1\}^{p(\lambda)}$	$\xrightarrow{r_0, r_1}$	Compute $\alpha\,|0, x_0, -\underbrace{(x_0 \cdot r_0)}_{z_0}\rangle + \beta\,|1, x_1, \underbrace{x_1 \cdot r_1}_{z_1}\rangle$
		$\mapsto \alpha\,|0, x_0\rangle + \omega_n^{z_0+z_1}\beta\,|1, x_1\rangle$

Blind RSP from TCFs: Construction I - Subprotocol($1^\lambda, n, |\psi\rangle$)

Client		**Server**
$((f_0, f_1), \mathrm{td}) \leftarrow \mathrm{Gen}(1^\lambda)$	$\xrightarrow{(f_0, f_1)}$	initial state: $\|\psi\rangle = \alpha\|0\rangle + \beta\|1\rangle$
	$\xleftarrow{y}$	Prepare $\alpha\|0, x_0\rangle + \beta\|1, x_1\rangle$
		where $f_0(x_0) = f_1(x_1) = y$
$r_0, r_1 \leftarrow \{0,1\}^{p(\lambda)}$	$\xrightarrow{r_0, r_1}$	Compute $\alpha\|0, x_0, -\underbrace{(x_0 \cdot r_0)}_{z_0}\rangle + \beta\|1, x_1, \underbrace{x_1 \cdot r_1}_{z_1}\rangle$
		$\mapsto \alpha\|0, x_0\rangle + \omega_n^{z_0+z_1}\beta\|1, x_1\rangle$
		Measure 2^{nd} register in Hadamard-basis:
$(x_0, x_1) \leftarrow \mathrm{Invert}(\mathrm{td}, y)$	$\xleftarrow{d}$	$d \in \{0,1\}^{p(\lambda)}, \|\psi'\rangle = \alpha\|0\rangle + \beta(-1)^{d\cdot(x_0 \oplus x_1)}\omega_n^{z_0+z_1}\|1\rangle$
$b := d \cdot (x_0 \oplus x_1)$		
$\theta := z_0 + z_1$		

Blind RSP from TCFs: Construction II - Protocol

$$\boxed{(b,\theta), \alpha|0\rangle + \beta(-1)^b\omega_n^\theta|1\rangle \leftarrow \mathsf{Subprotocol}(1^\lambda, n, \alpha|0\rangle + \beta|1\rangle)}$$

$$(b_1,\theta_1), |\psi_1\rangle \leftarrow \mathsf{Subprotocol}(1^\lambda, 2, |+\rangle)$$
$$(b_2,\theta_2), |\psi_2\rangle \leftarrow \mathsf{Subprotocol}(1^\lambda, 4, |\psi_1\rangle)$$
$$(b_3,\theta_3), |\psi_3\rangle \leftarrow \mathsf{Subprotocol}(1^\lambda, 8, |\psi_2\rangle)$$

$$b := b_1 \oplus b_2 \oplus b_3$$
$$\theta := 4\theta_1 + 2\theta_2 + \theta_3 \bmod 8$$
$$\textbf{return } (b, \theta \cdot \frac{\pi}{4}), |\psi_3\rangle$$

Blind RSP from TCFs: Correctness

$$\boxed{(b, \theta), \alpha |0\rangle + \beta(-1)^b \omega_n^\theta |1\rangle \leftarrow \text{Subprotocol}(1^\lambda, n, \alpha |0\rangle + \beta |1\rangle)}$$

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$(b_1, \theta_1), |\psi_1\rangle \leftarrow \text{Subprotocol}(1^\lambda, 2, |+\rangle)$

$(b_2, \theta_2), |\psi_2\rangle \leftarrow \text{Subprotocol}(1^\lambda, 4, |\psi_1\rangle)$

$(b_3, \theta_3), |\psi_3\rangle \leftarrow \text{Subprotocol}(1^\lambda, 8, |\psi_2\rangle)$

$b := b_1 \oplus b_2 \oplus b_3$

$\theta := (4\theta_1 + 2\theta_2 + \theta_3 \bmod 8)$

return $(b, \theta \cdot \frac{\pi}{4}), |\psi_3\rangle$

Blind RSP from TCFs: Correctness

$$\boxed{(b,\theta), \alpha|0\rangle + \beta(-1)^b\omega_n^\theta|1\rangle \leftarrow \mathsf{Subprotocol}(1^\lambda, n, \alpha|0\rangle + \beta|1\rangle)}$$

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$(b_1,\theta_1), |\psi_1\rangle \leftarrow \mathsf{Subprotocol}(1^\lambda, 2, |+\rangle) \qquad \mapsto \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{b_1}\omega_2^{\theta_1}|1\rangle)$$

$$(b_2,\theta_2), |\psi_2\rangle \leftarrow \mathsf{Subprotocol}(1^\lambda, 4, |\psi_1\rangle)$$

$$(b_3,\theta_3), |\psi_3\rangle \leftarrow \mathsf{Subprotocol}(1^\lambda, 8, |\psi_2\rangle)$$

$b := b_1 \oplus b_2 \oplus b_3$

$\theta := (4\theta_1 + 2\theta_2 + \theta_3 \bmod 8)$

return $(b, \theta \cdot \frac{\pi}{4}), |\psi_3\rangle$

Blind RSP from TCFs: Correctness

$$\boxed{(b,\theta), \alpha\,|0\rangle + \beta(-1)^b\omega_n^\theta\,|1\rangle \leftarrow \mathsf{Subprotocol}(1^\lambda, n, \alpha\,|0\rangle + \beta\,|1\rangle)}$$

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$(b_1,\theta_1), |\psi_1\rangle \leftarrow \mathsf{Subprotocol}(1^\lambda, 2, |+\rangle)$ $\quad\mapsto \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{b_1}\omega_2^{\theta_1}\,|1\rangle)$

$(b_2,\theta_2), |\psi_2\rangle \leftarrow \mathsf{Subprotocol}(1^\lambda, 4, |\psi_1\rangle)$

$(b_3,\theta_3), |\psi_3\rangle \leftarrow \mathsf{Subprotocol}(1^\lambda, 8, |\psi_2\rangle)$ $\quad\mapsto \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{b_1\oplus b_2}\omega_2^{\theta_1}\omega_4^{\theta_2}\,|1\rangle)$

$b := b_1 \oplus b_2 \oplus b_3$

$\theta := (4\theta_1 + 2\theta_2 + \theta_3 \bmod 8)$

return $(b, \theta\cdot\frac{\pi}{4}), |\psi_3\rangle$

Blind RSP from TCFs: Correctness

$$\boxed{(b, \theta), \alpha |0\rangle + \beta(-1)^b \omega_n^\theta |1\rangle \leftarrow \text{Subprotocol}(1^\lambda, n, \alpha |0\rangle + \beta |1\rangle)}$$

$(b_1, \theta_1), |\psi_1\rangle \leftarrow \text{Subprotocol}(1^\lambda, 2, |+\rangle)$

$(b_2, \theta_2), |\psi_2\rangle \leftarrow \text{Subprotocol}(1^\lambda, 4, |\psi_1\rangle)$

$(b_3, \theta_3), |\psi_3\rangle \leftarrow \text{Subprotocol}(1^\lambda, 8, |\psi_2\rangle)$

$b := b_1 \oplus b_2 \oplus b_3$

$\theta := (4\theta_1 + 2\theta_2 + \theta_3 \bmod 8)$

return $(b, \theta \cdot \frac{\pi}{4}), |\psi_3\rangle$

$$\begin{aligned}
&\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\
&\mapsto \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{b_1} \omega_2^{\theta_1} |1\rangle) \\
&\mapsto \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{b_1 \oplus b_2} \omega_2^{\theta_1} \omega_4^{\theta_2} |1\rangle) \\
&\mapsto \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{b_1 \oplus b_2 \oplus b_3} \omega_2^{\theta_1} \omega_4^{\theta_2} \omega_8^{\theta_3} |1\rangle) \\
&= \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{b_1 \oplus b_2 \oplus b_3} \omega_8^{4\theta_1 + 2\theta_2 + \theta_3} |1\rangle) \\
&= \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{b} \omega_8^{\theta} |1\rangle)
\end{aligned}$$

Blind RSP from TCFs: Blindness I

Theorem 3 (Quantum Goldreich-Levin)

If there exists a quantum algorithm that given a random r and an auxiliary quantum input ρ_x computes $r \cdot x$ with probability at least $1/2 + \varepsilon$, then there exists a quantum algorithm that takes ρ_x and extracts x with probability $4\varepsilon^2$.

$$z_0 \oplus z_1 = (x_0 \cdot r_0) \oplus (x_1 \cdot r_1) = (x_0 || x_1) \cdot (r_0 || r_1) = r \cdot x$$

Blind RSP from TCFs: Blindness I

Theorem 3 (Quantum Goldreich-Levin)

If there exists a quantum algorithm that given a random r and an auxiliary quantum input ρ_x computes $r \cdot x$ with probability at least $1/2 + \varepsilon$, then there exists a quantum algorithm that takes ρ_x and extracts x with probability $4\varepsilon^2$.

$$z_0 \oplus z_1 = (x_0 \cdot r_0) \oplus (x_1 \cdot r_1) = (x_0 || x_1) \cdot (r_0 || r_1) = r \cdot x$$

If $(\mathrm{Gen}, \mathrm{Invert})$ is a TCF, then $z_0 \oplus z_1 \approx_c z^*$ where $z^* \leftarrow \{0,1\}$.

Blind RSP from TCFs: Blindness II

$$\begin{aligned}\theta &= 4\theta_1 + 2\theta_2 + \theta_3 \text{ mod } 8 \\ &= 4(z_{1,0} + z_{1,1}) + 2(z_{2,0} + z_{2,1}) + (z_{3,0} + z_{3,1}) \text{ mod } 8 \\ &= 4(z_{1,0} + z_{1,1}) + 2(z_{1,0} + z_{1,1} + \tilde{z}_3) + (z_{3,0} \oplus z_{3,1}) \text{ mod } 8 \\ &= 4(z_{1,0} + z_{1,1} + \tilde{z}_2) + 2(z_{1,0} \oplus z_{1,1} \oplus \tilde{z}_3) + (z_{3,0} \oplus z_{3,1}) \text{ mod } 8 \\ &= 4\underbrace{(z_{1,0} \oplus z_{1,1} \oplus \tilde{z}_2)}_{\theta'_1} + 2\underbrace{(z_{2,0} \oplus z_{2,1} \oplus \tilde{z}_3)}_{\theta'_2} + \underbrace{(z_{3,0} \oplus z_{3,1})}_{\theta'_3}\end{aligned}$$

Blind RSP from TCFs: Blindness II

$$\begin{aligned}
\theta &= 4\theta_1 + 2\theta_2 + \theta_3 \bmod 8 \\
&= 4(z_{1,0} + z_{1,1}) + 2(z_{2,0} + z_{2,1}) + (z_{3,0} + z_{3,1}) \bmod 8 \\
&= 4(z_{1,0} + z_{1,1}) + 2(z_{1,0} + z_{1,1} + \tilde{z}_3) + (z_{3,0} \oplus z_{3,1}) \bmod 8 \\
&= 4(z_{1,0} + z_{1,1} + \tilde{z}_2) + 2(z_{1,0} \oplus z_{1,1} \oplus \tilde{z}_3) + (z_{3,0} \oplus z_{3,1}) \bmod 8 \\
&= 4\underbrace{(z_{1,0} \oplus z_{1,1} \oplus \tilde{z}_2)}_{\theta'_1} + 2\underbrace{(z_{2,0} \oplus z_{2,1} \oplus \tilde{z}_3)}_{\theta'_2} + \underbrace{(z_{3,0} \oplus z_{3,1})}_{\theta'_3}
\end{aligned}$$

If $(\mathsf{Gen}, \mathsf{Invert})$ is a TCF, then

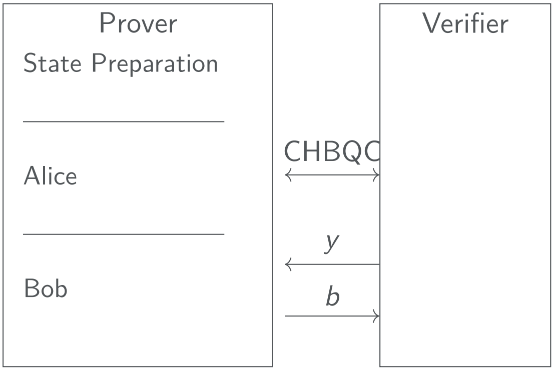
$$4\theta'_1 + 2\theta'_2 + \theta'_3 \approx_c 4\theta^*_1 + 2\theta'_2 + \theta'_3 \approx_c \cdots \approx_c 4\theta^*_1 + 2\theta^*_2 + \theta^*_3 = \theta^*$$

where $\theta^*_i \leftarrow \{0, 1\}, \theta^* \leftarrow \mathbb{Z}_8$. Thus $\theta \cdot \frac{\pi}{4}$ indistinguishable from $\alpha \leftarrow \{0, \ldots, \frac{7\pi}{4}\}$.

Table of Contents

New Compiler

Half-Blind Quantum Computing + Blind Remote State Preparation = CHBQC



Comparison

	KLVY	this work
Assumption	QHE	any TCF
Round complexity	constant	linear in Alice's circuit
Quantum Completeness	✓	✓
Quantum Soundness	✓	✓

Application: CVQC from any TCF

- CVQC from LWE using Hamiltonian-based approach [Mahadev]
- CVQC from QHE using compiled CHSH nonlocal game [Natarajan-Zhang]

Application: CVQC from any TCF

- CVQC from LWE using Hamiltonian-based approach [Mahadev]
- CVQC from QHE using compiled CHSH nonlocal game [Natarajan-Zhang]

Adapting the proofs of [Natarajan-Zhang] to new compiler
$\Rightarrow$ CVQC from any TCF

Open Questions

- Compiler from any TCF with constant round complexity?
- Other applications for blind RSP protocol from any TCF?

Thank you!

ePrint:2024/1829, accepted at CRYPTO'25