Encrypted Quantum Computation:
Cryptography for the Quantum Cloud

James Bartusek

UC Berkeley

NEWS | 23 October 2019

Hello quantum world! Google publishes landmark quantum supremacy claim

The company says that its quantum computer is the first to perform a calculation that would be practically impossible for a classical machine.

How can we verify these claims?

How can we "prove quantumness" to classical machines?

TECH

IBM and Google disagree on quantum computing achievement

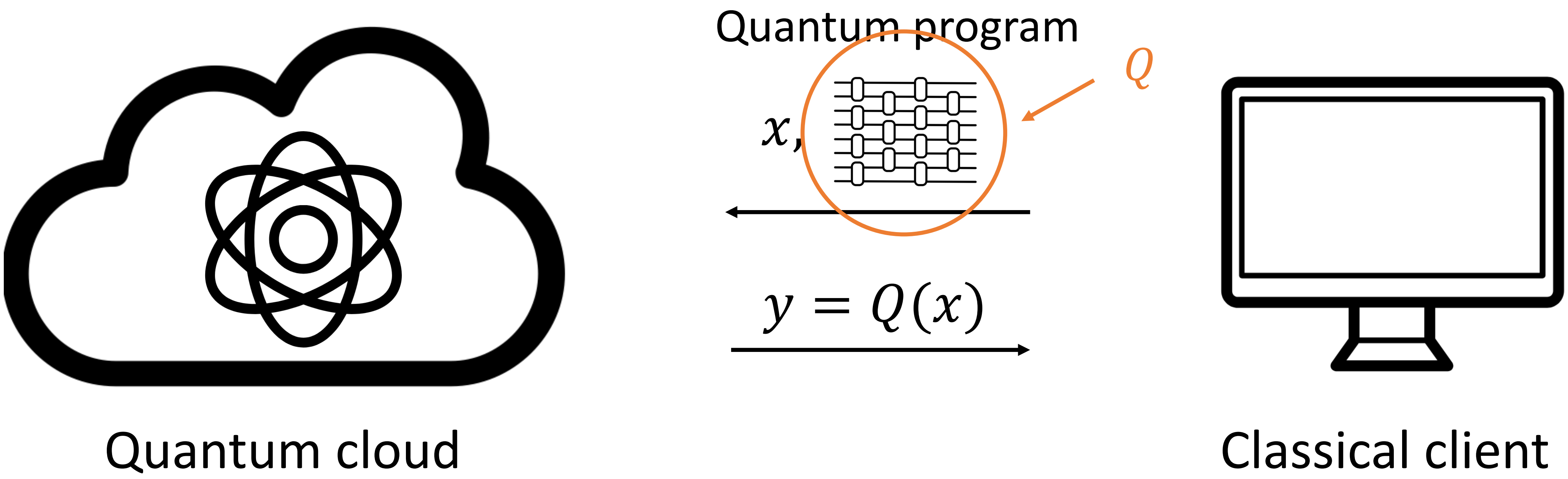
PUBLISHED WED, OCT 23 2019•2:16 PM EDT | UPDATED WED, OCT 23 2019•4:00 PM EDT

Suppose we have managed to achieve “quantum advantage”

Another issue: Building quantum computers is extremely costly

In the near-term, quantum computing technology will be highly concentrated

Delegation of Quantum Computation



Desirable security properties:

- Blindness: the cloud learns nothing about the client's input x
- Verifiability: the client can be sure that the output y is computed correctly

The Plan

- Part 1: Quantum background
- Part 2: Blind delegation from oblivious state preparation
- Part 3: Oblivious state preparation from post-quantum crypto
- Part 4: Proofs of quantumness and verifiable delegation

Part 1: Quantum Background

- How to encrypt quantum states
- Quantum universal gate set

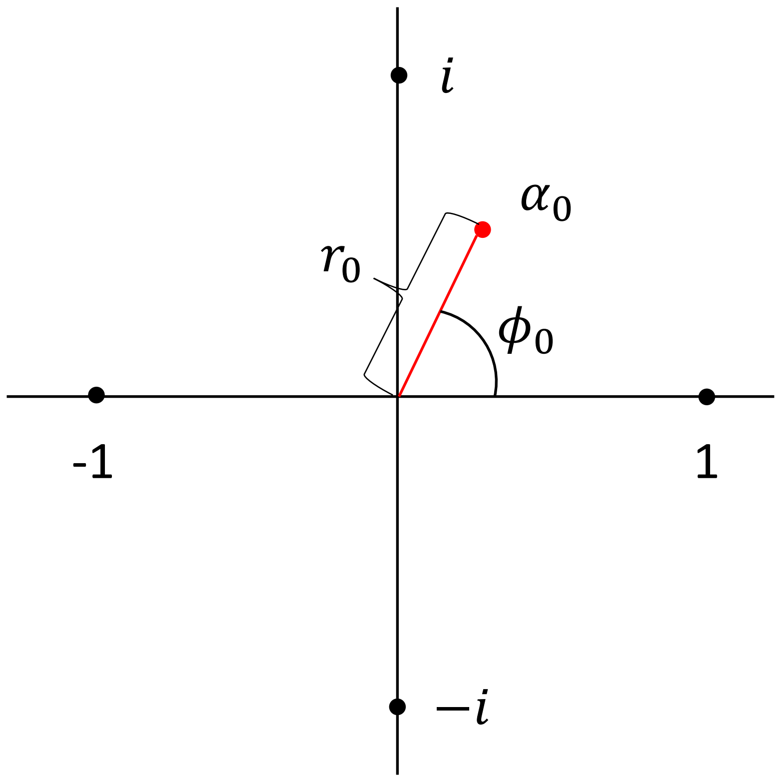
The Bloch Sphere

Single-qubit state:

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle \quad (\alpha_0, \alpha_1 \in \mathbb{C}, |\alpha_0|^2 + |\alpha_1|^2 = 1)$$

$$= r_0 e^{i\phi_0}|0\rangle + r_1 e^{i\phi_1}|1\rangle, \quad \phi_0, \phi_1 \in [0, 2\pi)$$

$$= e^{i\phi_0}(r_0|0\rangle + r_1 e^{i(\phi_1 - \phi_0)}|1\rangle)$$



The Bloch Sphere

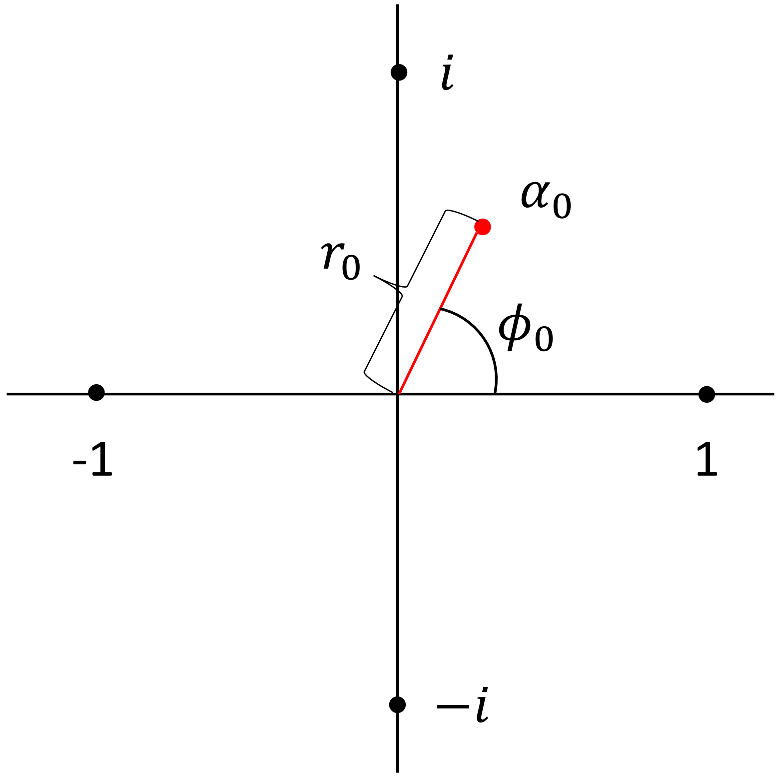
Single-qubit state:

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle \ \ (\alpha_0, \alpha_1 \in \mathbb{C}, |\alpha_0|^2 + |\alpha_1|^2 = 1)$$

$$= r_0 e^{i\phi_0}|0\rangle + r_1 e^{i\phi_1}|1\rangle, \quad \phi_0, \phi_1 \in [0,2\pi)$$

$$= r_0|0\rangle + r_1 e^{i\phi}|1\rangle, \quad \phi \in [0,2\pi)$$

$$= \cos(\theta/2)|0\rangle + \sin(\theta/2)e^{i\phi}|1\rangle, \ \theta \in [0,\pi]$$



The Bloch Sphere

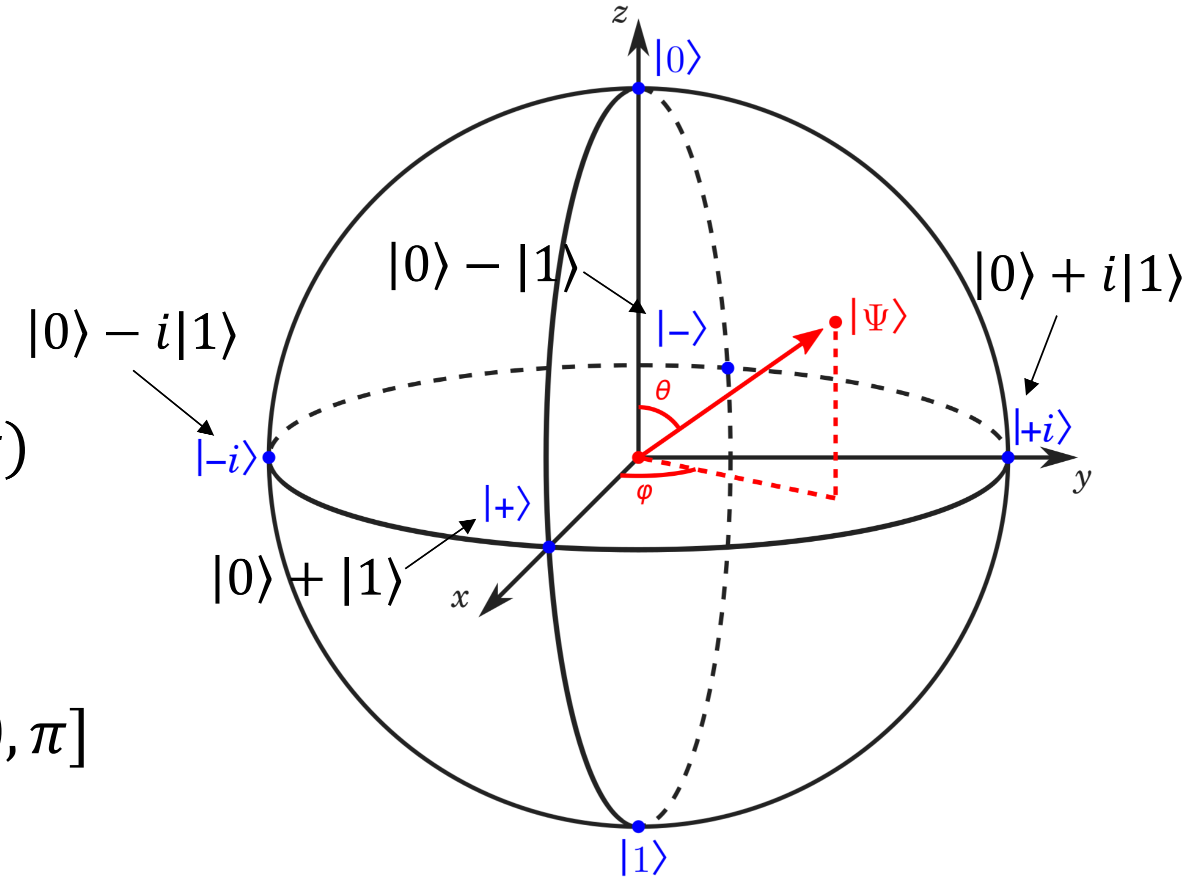
Single-qubit state:

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle \quad (\alpha_0, \alpha_1 \in \mathbb{C}, |\alpha_0|^2 + |\alpha_1|^2 = 1)$$

$$= r_0 e^{i\phi_0}|0\rangle + r_1 e^{i\phi_1}|1\rangle, \quad \phi_0, \phi_1 \in [0,2\pi)$$

$$= r_0|0\rangle + r_1 e^{i\phi}|1\rangle, \quad \phi \in [0,2\pi)$$

$$= \cos(\theta/2)|0\rangle + \sin(\theta/2)e^{i\phi}|1\rangle, \quad \theta \in [0,\pi]$$



Convention: drop normalization factors when clear from context

The Bloch Sphere

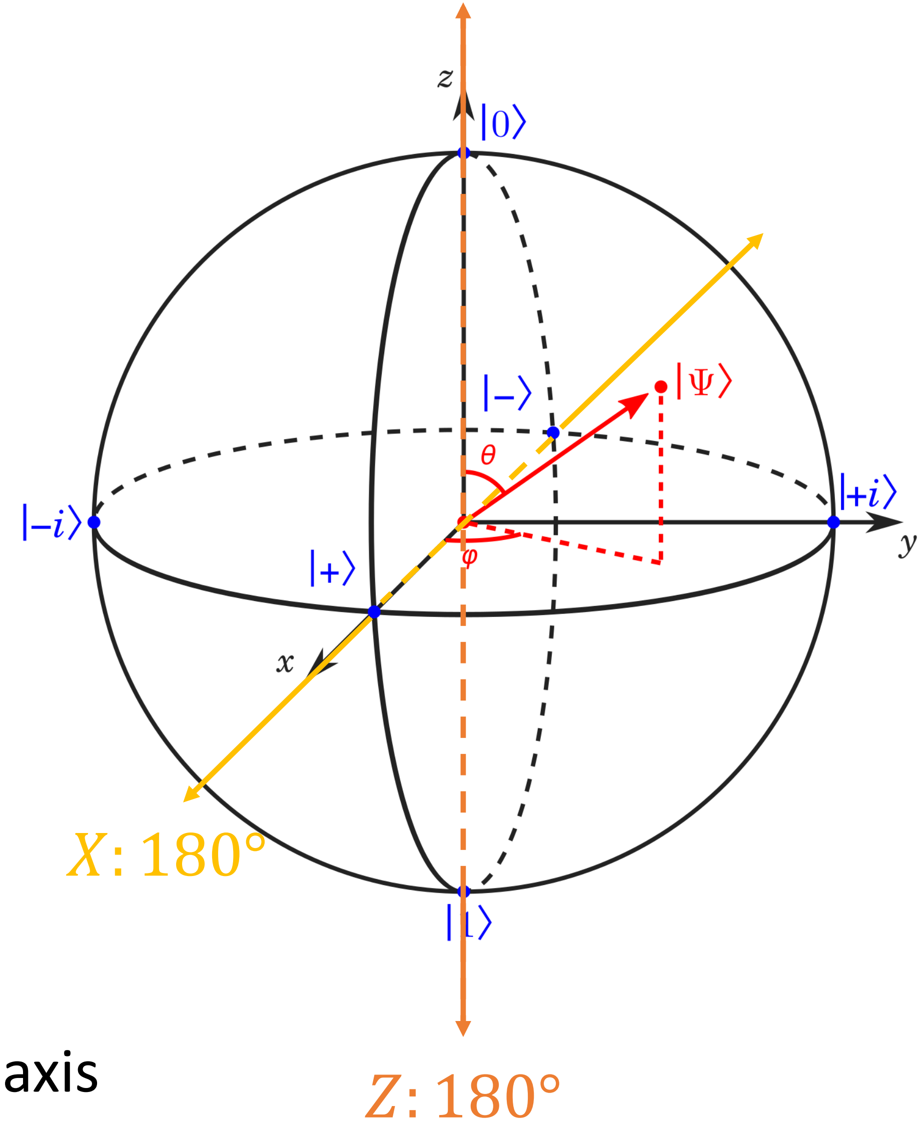
- Any single-qubit state can be represented as a point on the unit sphere

- Any single-qubit unitary can be represented as a rotation of the unit sphere

- Pauli rotations:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$ "bit flip": 180° around the x-axis

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$ "phase flip": 180° around the z-axis

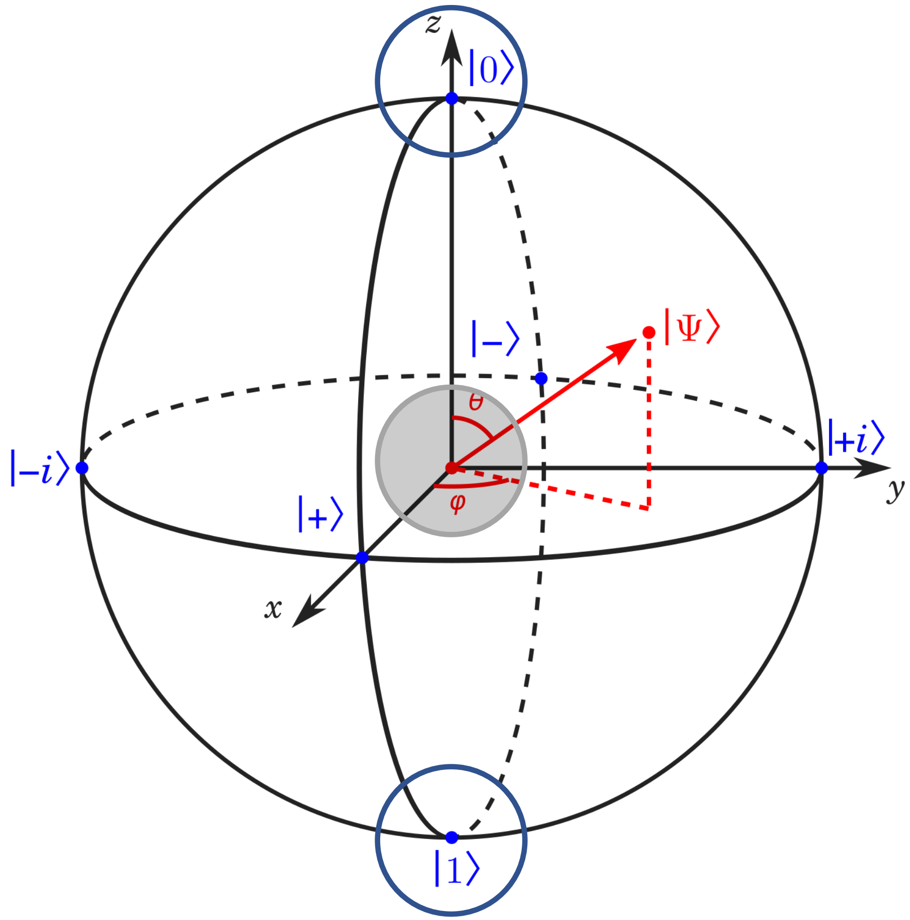


How to encrypt quantum states

Classical one-time pad:

To encrypt a bit b, sample random $r \leftarrow \{0,1\}$, and output $b \oplus r$ $\quad (= X^r|b\rangle)$

"encrypting θ"



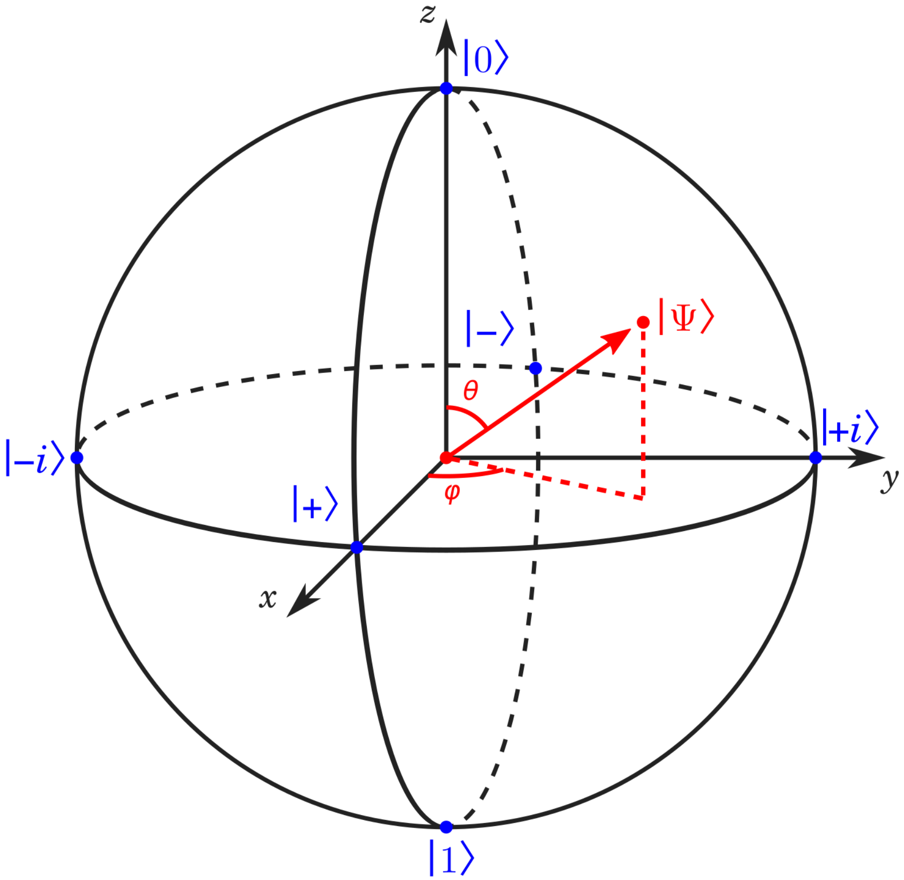
How to encrypt quantum states

Classical one-time pad:

To encrypt a bit b, sample random $r \leftarrow \{0,1\}$, and output $b \oplus r$ $\quad (= X^r|b\rangle)$

Quantum one-time pad [MTdW00]:

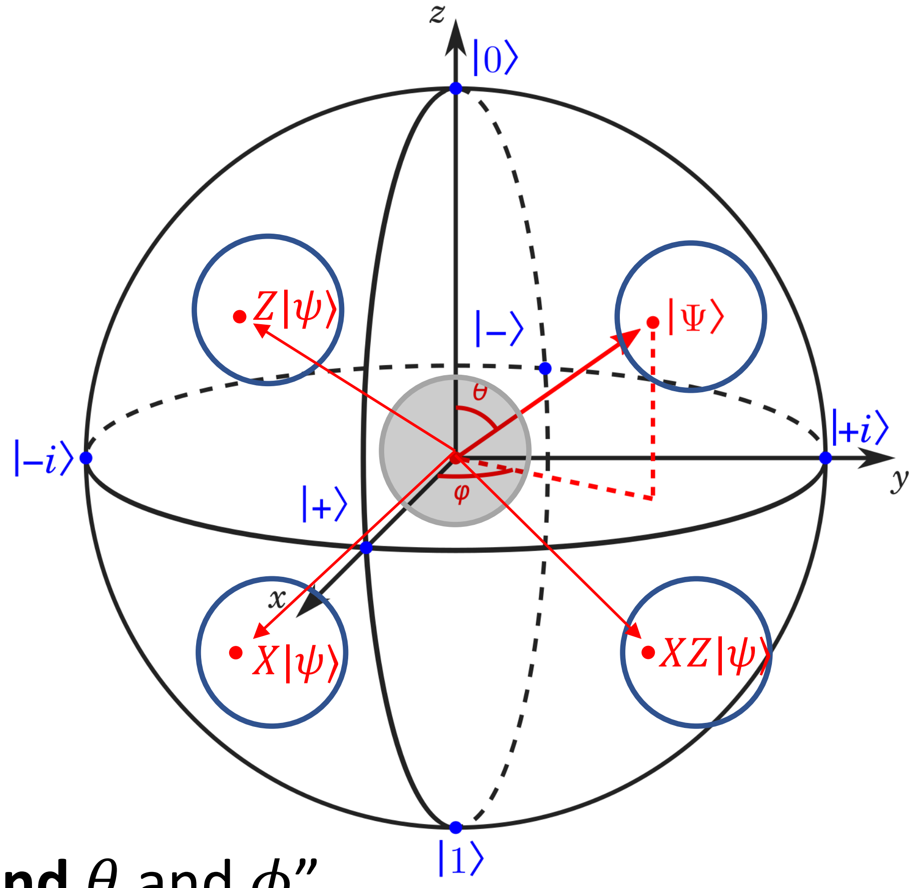
To encrypt a state $|\psi\rangle$, sample random $r, s \leftarrow \{0,1\}$, and output $X^r Z^s |\psi\rangle$



How to encrypt quantum states

Classical one-time pad:

To encrypt a bit b, sample random $r \leftarrow \{0,1\}$, and output $b \oplus r$ $\quad (= X^r|b\rangle)$



Quantum one-time pad [MTdW00]:

To encrypt a state $|\psi\rangle$, sample random $r, s \leftarrow \{0,1\}$, and output $X^r Z^s |\psi\rangle$

"encrypting **and** θ and ϕ"

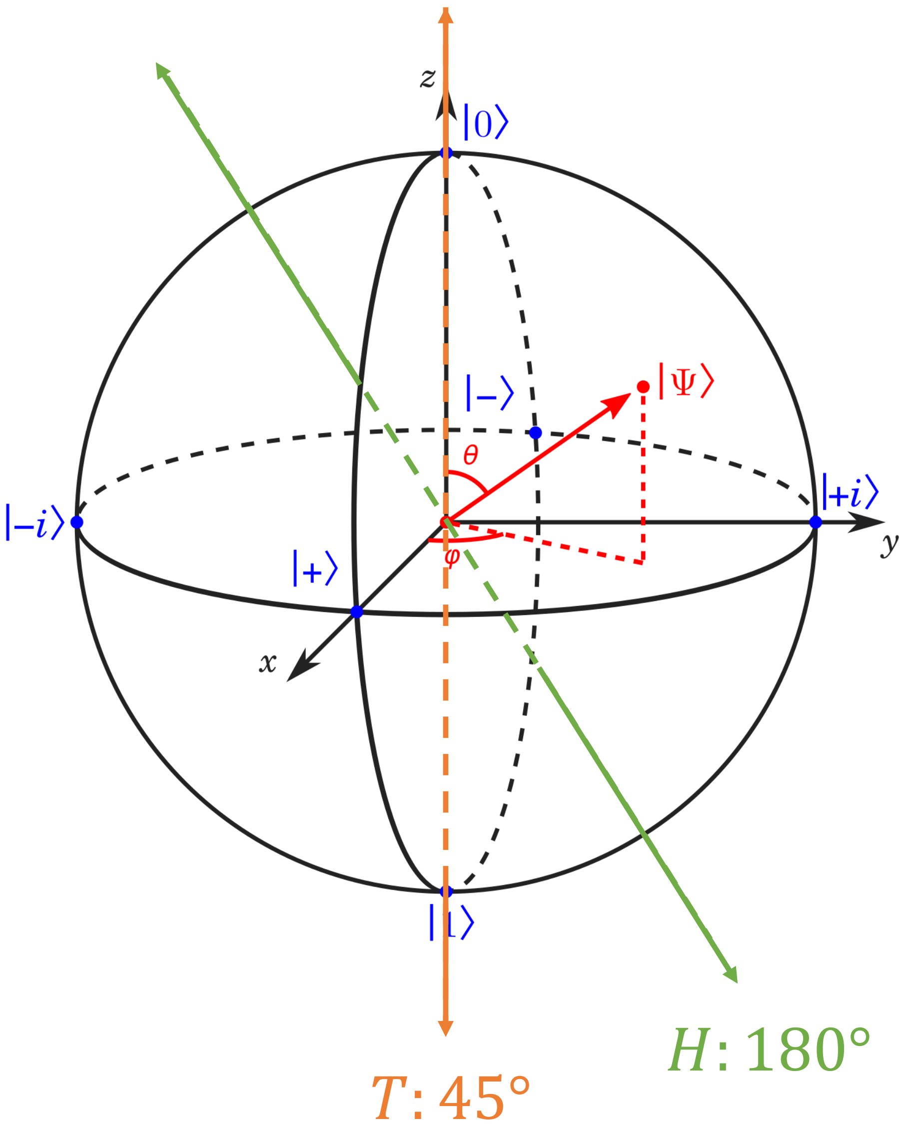
Extends to n-qubit states:

Sample $r, s \leftarrow \{0,1\}^n$, and output $X^{r_1} Z^{s_1} \otimes \cdots \otimes X^{r_n} Z^{s_n} |\psi\rangle \coloneqq X^r Z^s |\psi\rangle$

Universal gate set

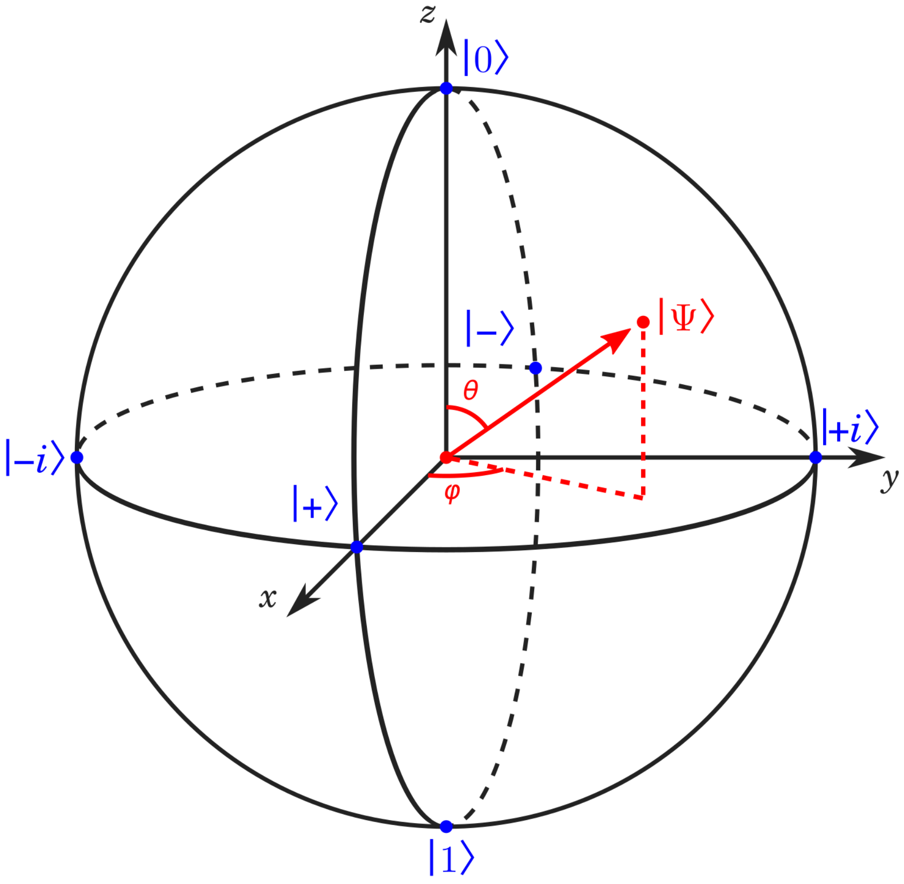
- Consider any n-qubit unitary U
- Goal: write U (approximately) as a sequence of one- and two-qubit gates, from a small finite set
- Claim #1: Any U can be written as a series of single-qubit rotations and CNOT gates, where CNOT: $|x\rangle|y\rangle \rightarrow |x\rangle|x \oplus y\rangle$
- Claim #2: Any single-qubit rotation can be written (approximately) as a series of:
  - Hadamard gate $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
  - T gate $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$
- Claim #3 (Solovay-Kitaev): This approximation is efficient

(A good reference for all of these claims is Nielsen-Chuang)



Clifford gates

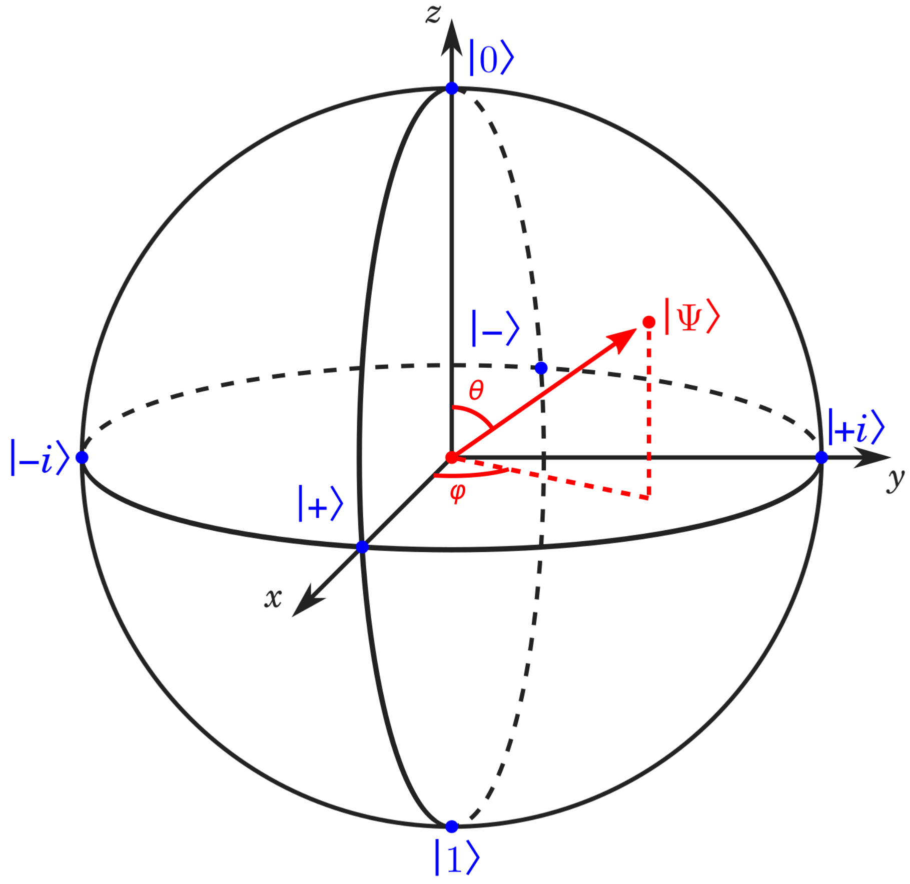
- Recall: QOTP $X^r Z^s |\psi\rangle, r, s, \in \{0,1\}^n$
- Clifford group normalizes the Pauli group
  - For any Clifford gate C, $CX^r Z^s = X^{r'} Z^{s'} C$



Clifford gates

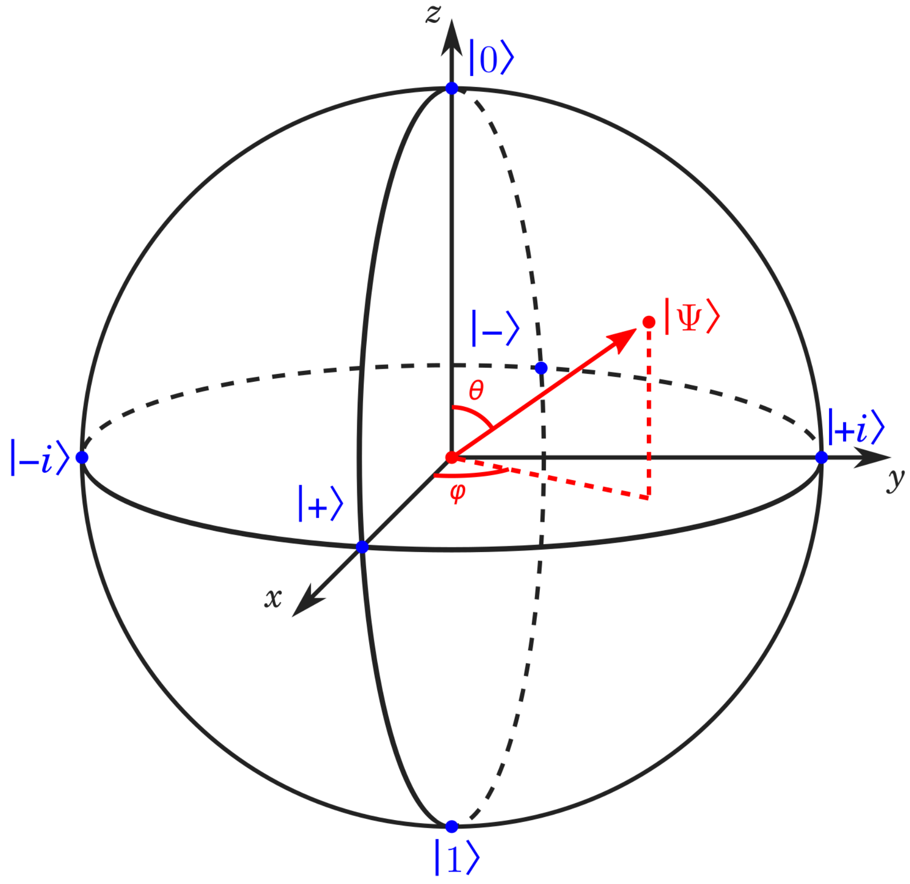
- Recall: QOTP $X^r Z^s|\psi\rangle, r, s, \in \{0,1\}^n$
- Clifford group normalizes the Pauli group
  - For any Clifford gate C, $CX^r Z^s|\psi\rangle = X^{r'} Z^{s'} C|\psi\rangle$
  - Cliffords can be applied directly to encrypted quantum states, and the QOTP key get updated
- Recall: Universal gate set CNOT, H, T
- CNOT is Clifford: $\mathrm{CNOT}(X^{r_1}Z^{s_1} \otimes X^{r_2}Z^{s_2})$
  $= \left(X^{r_1}Z^{s_1 \oplus s_2} \otimes X^{r_1 \oplus r_2}Z^{s_2}\right)\mathrm{CNOT}$
- H is Clifford: $HX^r Z^s = X^s Z^r H$
- T is **not** Clifford:

$$TX = T^2XT$$



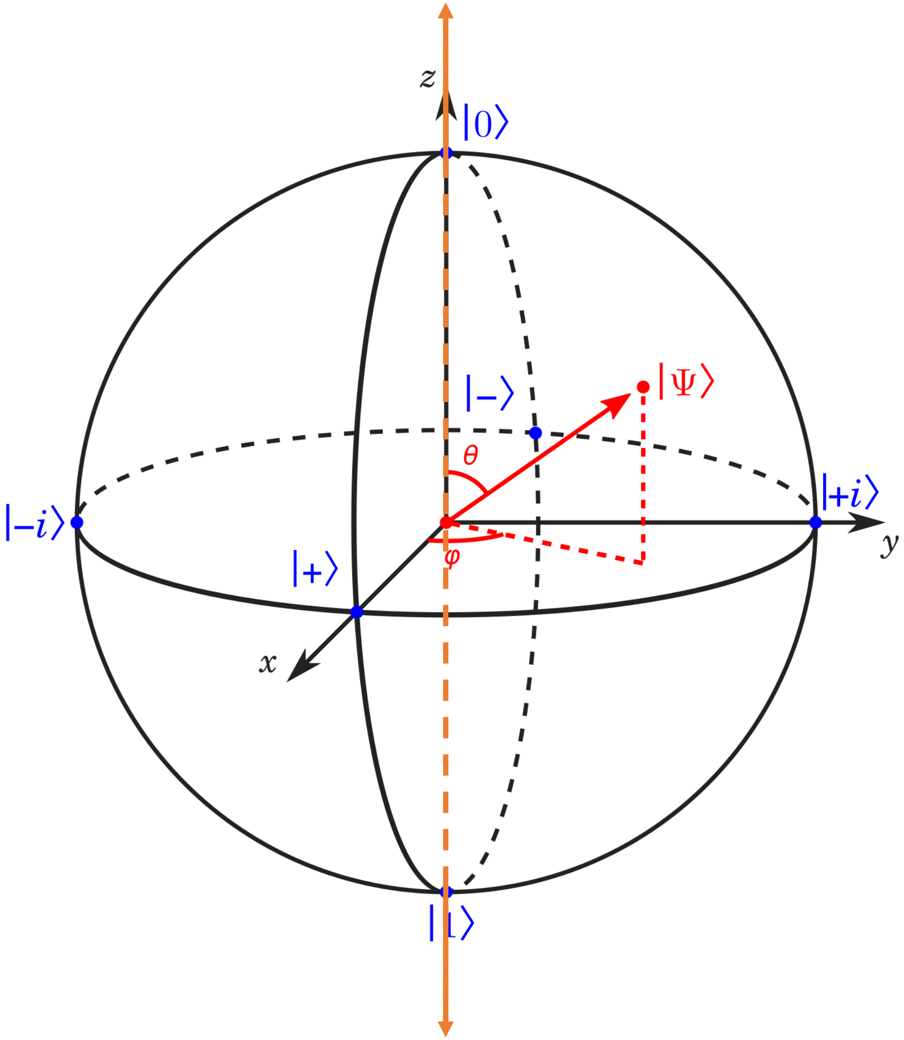
Clifford gates

- Recall: QOTP $X^r Z^s |\psi\rangle, r, s, \in \{0,1\}^n$
- Clifford group normalizes the Pauli group
  - For any Clifford gate C, $CX^r Z^s |\psi\rangle = X^{r'} Z^{s'} C |\psi\rangle$
  - Cliffords can be applied directly to encrypted quantum states, and the QOTP key get updated
- Recall: Universal gate set CNOT, H, T
- CNOT is Clifford: $\text{CNOT}(X^{r_1} Z^{s_1} \otimes X^{r_2} Z^{s_2})$
  $= \left(X^{r_1} Z^{s_1 \oplus s_2} \otimes X^{r_1 \oplus r_2} Z^{s_2}\right)\text{CNOT}$
- H is Clifford: $HX^r Z^s = X^s Z^r H$
- T is **not** Clifford: $TX^r Z^s = (T^2)^r X^r Z^s T$



Clifford gates

- Recall: QOTP $X^r Z^s |\psi\rangle, r, s, \in \{0,1\}^n$
- Clifford group normalizes the Pauli group
  - For any Clifford gate C, $CX^r Z^s |\psi\rangle = X^{r'} Z^{s'} C|\psi\rangle$
  - Cliffords can be applied directly to encrypted quantum states, and the QOTP key get updated
- Recall: Universal gate set CNOT, H, T
- CNOT is Clifford: $\mathrm{CNOT}(X^{r_1} Z^{s_1} \otimes X^{r_2} Z^{s_2})$
  $= \left(X^{r_1} Z^{s_1 \oplus s_2} \otimes X^{r_1 \oplus r_2} Z^{s_2}\right)\mathrm{CNOT}$
- H is Clifford: $HX^r Z^s = X^s Z^r H$
- T is **not** Clifford: $TX^r Z^s = P^r X^r Z^s T$
- P is called the "phase gate"



P: 90°

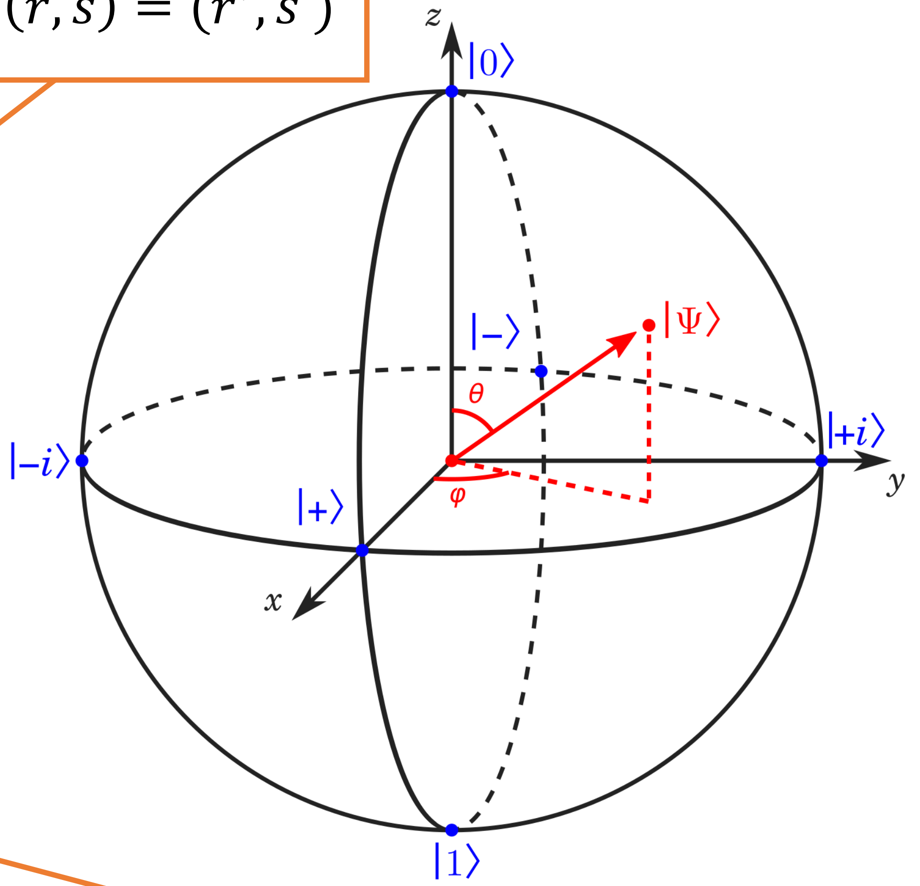
Recap

Key property: For any Clifford C and $r, s \in \{0,1\}^n$, there exists $r', s' \in \{0,1\}^n$ such that $CX^rZ^s = X^{r'}Z^{s'}C$

Define f_C to be the "update function": $f_C(r,s) = (r', s')$

- How to encrypt quantum states: $X^rZ^s|\psi\rangle$
- Universal gate set: CNOT, H, T
- CNOT and H are Clifford gates
- Any quantum computation Q can be performed using just Clifford computations and T gates
- $TX^rZ^s = P^rX^rZ^sT$

That is, $Q(x) = C_tTC_{t-1} \ldots TC_2TC_1|x\rangle$

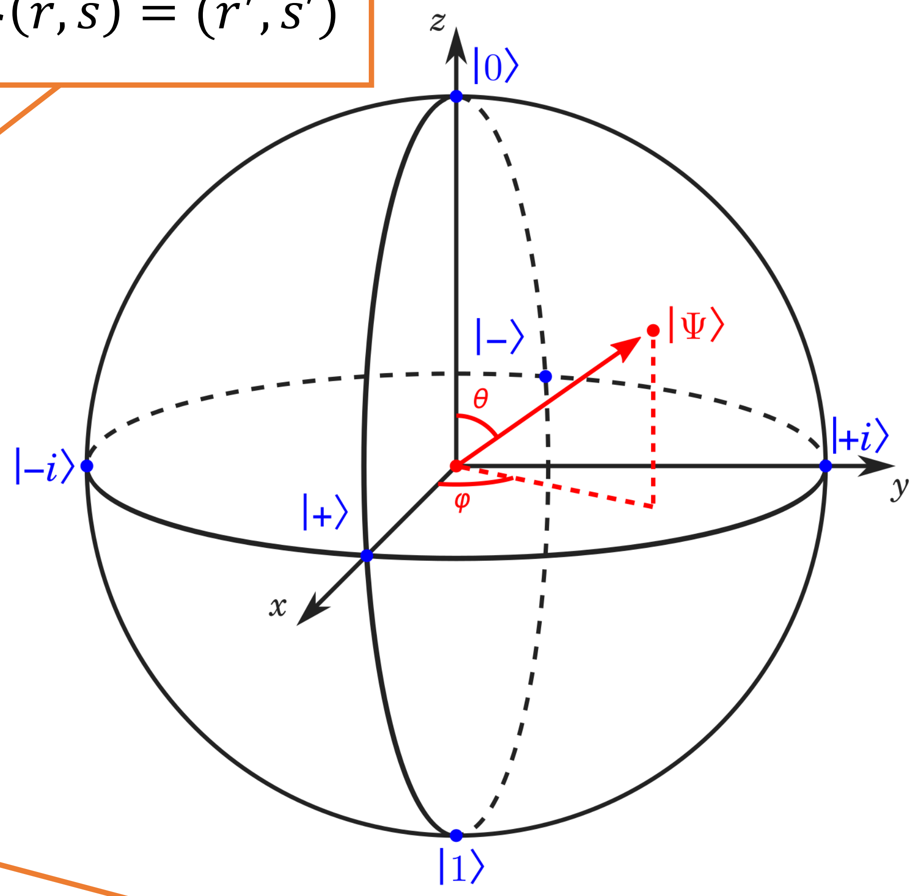


Recap

Key property: For any Clifford C and $r, s \in \{0,1\}^n$, there exists $r', s' \in \{0,1\}^n$ such that $CX^rZ^s = X^{r'}Z^{s'}C$

Define f_C to be the "update function": $f_C(r,s) = (r', s')$

- How to encrypt quantum states: $X^rZ^s|\psi\rangle$
- Universal gate set: CNOT, H, T
- CNOT and H are Clifford gates
- Any quantum computation Q can be performed using just Clifford computations and T gates
- $T^\dagger X^r Z^s = \left(P^\dagger\right)^r X^r Z^s T^\dagger$



That is, $Q(x) = C_t T^\dagger C_{t-1} \ldots T^\dagger C_2 T^\dagger C_1 |x\rangle$

Part 2: Blind Delegation from Oblivious BB84 State Preparation

Quantum server $\quad Q = C_t T^\dagger C_{t-1} \ldots T^\dagger C_2 T^\dagger C_1 \quad$ Classical client(x)

Sample $r \leftarrow \{0,1\}^n$

Initialize $(r_0, s_0) = (r, 0^n)$

$\xleftarrow{\quad r_0 \oplus x \quad}$

Initialize $|\psi_0\rangle = |r_0 \oplus x\rangle = X^{r_0} Z^{s_0} |x\rangle$

Quantum server $Q = (C_t)(T^\dagger C_{t-1}) \dots (T^\dagger C_2)(T^\dagger C_1)$ Classical client (x)

Sample $r \leftarrow \{0,1\}^n$

$r_0 \oplus x$ ←

Initialize $|\psi_0\rangle = |r_0 \oplus x\rangle = X^{r_0} Z^{s_0} |x\rangle$

Initialize $(r_0, s_0) = (r, 0^n)$

Compute $|\psi_1\rangle = T^\dagger C_1 |\psi_0\rangle = T^\dagger X^{r_1} Z^{s_1} C_1 |x\rangle$

Update $(r_1, s_1) = f_{C_1}(r_0, s_0)$

Quantum server $Q = (C_t)(T^\dagger C_{t-1}) \dots (T^\dagger C_2)(T^\dagger C_1)$ **Classical client** (x)

Sample $r \leftarrow \{0,1\}^n$

$r_0 \oplus x$ ←

Initialize $|\psi_0\rangle = |r_0 \oplus x\rangle = X^{r_0} Z^{s_0} |x\rangle$

Initialize $(r_0, s_0) = (r, 0^n)$

Compute $|\psi_1\rangle = T^\dagger C_1 |\psi_0\rangle = (P^\dagger)^{r_{1,1}} X^{r_1} Z^{s_1} T^\dagger C_1 |x\rangle$

Update $(r_1, s_1) = f_{C_1}(r_0, s_0)$

$|\psi_1\rangle$ →

Oblivious phase correction

← $r_{1,1}$

$X^{r_1} Z^{s_1} T^\dagger C_1 |x\rangle = P^{r_{1,1}} |\psi_1\rangle$ ←

Quantum server $Q = (C_t)(T^\dagger C_{t-1}) \ldots (T^\dagger C_2)(T^\dagger C_1)$ **Classical client** (x)

Sample $r \leftarrow \{0,1\}^n$

Initialize $|\psi_0\rangle = |r_0 \oplus x\rangle = X^{r_0} Z^{s_0} |x\rangle$ $\xleftarrow{\; r_0 \oplus x \;}$ Initialize $(r_0, s_0) = (r, 0^n)$

Compute $|\psi_1\rangle = T^\dagger C_1 |\psi_0\rangle = (P^\dagger)^{r_{1,1}} X^{r_1} Z^{s_1} T^\dagger C_1 |x\rangle$ Update $(r_1, s_1) = f_{C_1}(r_0, s_0)$

$|\psi_1\rangle \rightarrow$ **Oblivious phase correction** $\leftarrow r_{1,1}$

$|\psi_1'\rangle = X^{r_1} Z^{s_1} T^\dagger C_1 |x\rangle = P^{r_{1,1}} |\psi_1\rangle \leftarrow$

Compute $|\psi_2\rangle = T^\dagger C_2 |\psi_1'\rangle = (P^\dagger)^{r_{2,1}} X^{r_2} Z^{s_2} T^\dagger C_2 T^\dagger C_1 |x\rangle$ Update $(r_2, s_2) = f_{C_2}(r_1, s_1)$

$|\psi_2\rangle \rightarrow$ **Oblivious phase correction** $\leftarrow r_{2,1}$

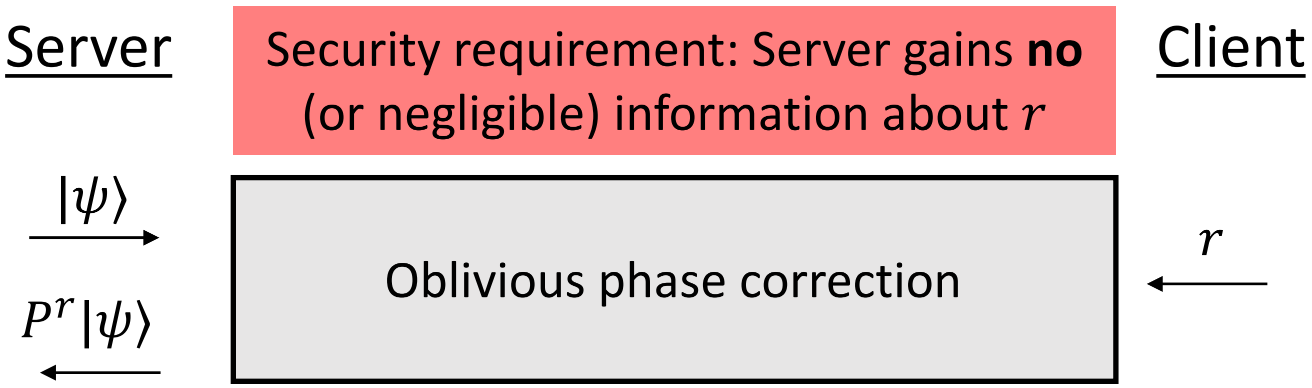
$|\psi_2'\rangle = P^{r_{2,1}} |\psi_2\rangle \leftarrow$

⋮

Compute $|\psi_t\rangle = C_t |\psi_{t-1}'\rangle$

$= X^{r_t} Z^{s_t} C_t T^\dagger C_{t-1} \ldots T^\dagger C_1 |x\rangle$ Update $(r_t, s_t) = f_{C_t}(r_{t-1}, s_{t-1})$

$= X^{r_t} Z^{s_t} |Q(x)\rangle = |r_t \oplus Q(x)\rangle$ $\xrightarrow{\; r_t \oplus Q(x) \;}$ Recover $Q(x)$



- The previous protocol template was first developed by Childs in 2001
  - Implemented oblivious phase correction using two-way quantum communication

- This was improved by Broadbent in 2015 to one-way quantum communication

- In 2017, Mahadev introduced techniques that allow us to implement oblivious phase correction with only classical communication

Oblivious Phase via Oblivious State Preparation

Recall: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \xrightarrow{P} |\psi\rangle = \alpha|0\rangle + i\beta|1\rangle$

"Magic state" based implementation:

Only difference

1. Prepare resource state $|0\rangle + i|1\rangle$
2. Compute $\text{CNOT}|\psi\rangle(|0\rangle + i|1\rangle)$

   $= \alpha|00\rangle + i\alpha|01\rangle + \beta|11\rangle + i\beta|10\rangle$
3. Measure 2nd qubit $\rightarrow m \in \{0,1\}$:

   If $m = 0$: $\alpha|0\rangle + i\beta|1\rangle$ If $m = 1$: $i\alpha|0\rangle + \beta|1\rangle$

   $= \alpha|0\rangle - i\beta|1\rangle$

   $= Z(\alpha|0\rangle + i\beta|1\rangle)$

Result: $Z^m P|\psi\rangle$

1. Prepare resource state $|0\rangle + |1\rangle$
2. Compute $\text{CNOT}|\psi\rangle(|0\rangle + |1\rangle)$

   $= \alpha|00\rangle + \alpha|01\rangle + \beta|11\rangle + \beta|10\rangle$
3. Measure 2nd qubit $\rightarrow m \in \{0,1\}$:

   If $m = 0$: $\alpha|0\rangle + \beta|1\rangle$ If $m = 1$: $\alpha|0\rangle + \beta|1\rangle$

Result: $|\psi\rangle$

Oblivious Phase via Oblivious State Preparation

Server | **Client**

Security requirement: Server gains **no** (or negligible) information about r

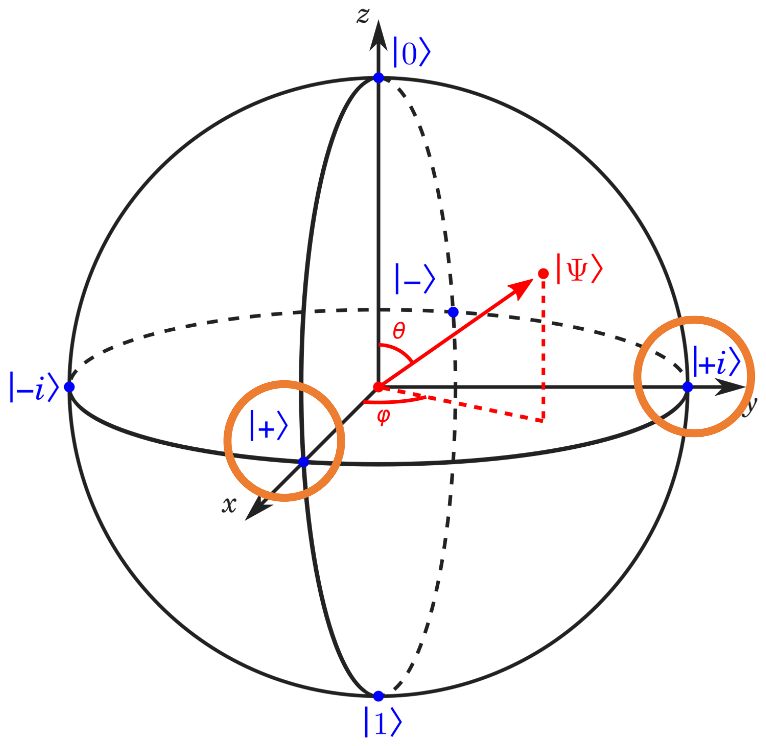
Server → $|\psi\rangle$; Server ← $P^r|\psi\rangle$

Oblivious phase correction ← r (Client)

↓ Reduces to

Server ← $|0\rangle + |1\rangle$ if $r = 0$; $|0\rangle + i|1\rangle$ if $r = 1$

Oblivious state preparation ← r (Client)



Oblivious Phase via Oblivious State Preparation

Server

Security requirement: Server gains **no** (or negligible) information about r

Client

$|\psi\rangle$ →

$P^r|\psi\rangle$ ←

Oblivious phase correction

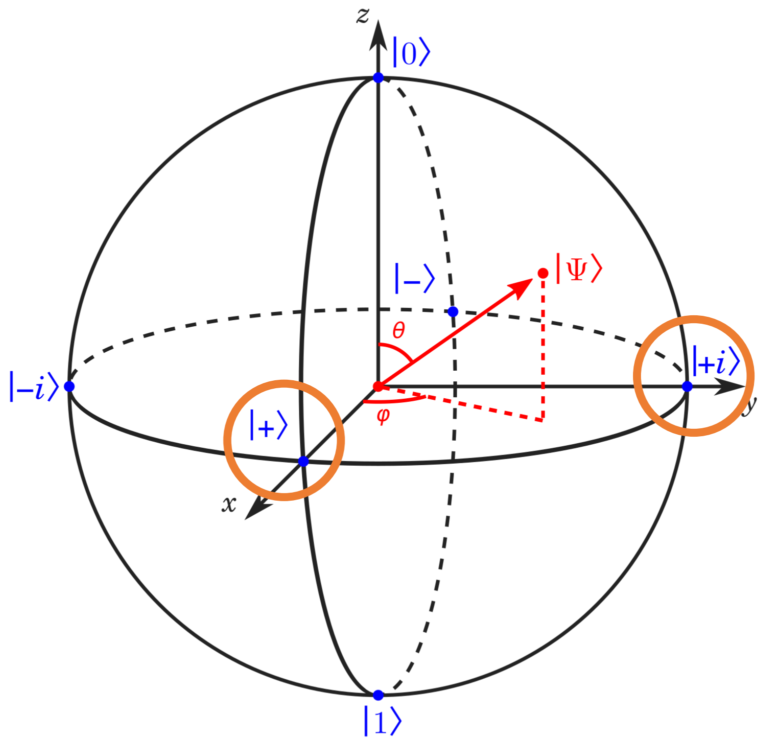
← r

Reduces to

$P^r|+\rangle$ ←

Oblivious state preparation

← r



As stated, no protocol can achieve the security requirement:

- Suppose Server measures received state in the $\{|+\rangle, |-\rangle\}$ basis
- If $r = 0$, the Server will see $|+\rangle$ with probability 1
- If $r = 1$, the Server will see $|+\rangle$ or $|-\rangle$ each with probability ½

Oblivious Phase via Oblivious State Preparation

Server | **Client**

Security requirement: Server gains **no** (or negligible) information about r

$|\psi\rangle$ →

← $Z^b P^r |\psi_1\rangle$

Oblivious phase correction

← r

b →

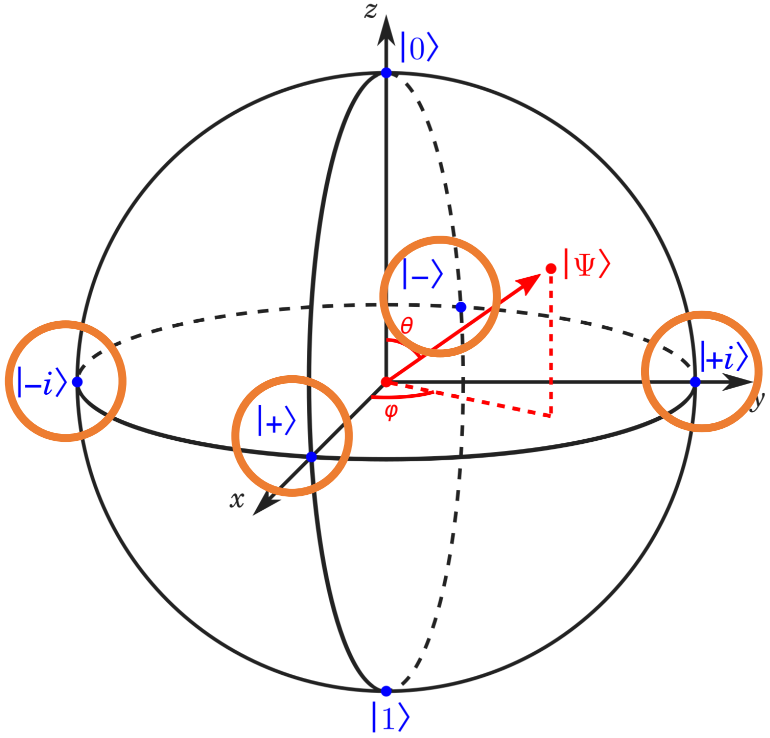
↓ Reduces to

← $Z^b P^r |+\rangle$

Oblivious state preparation

← r

b →



Solution: Allow for potential phase flip

Oblivious Phase via Oblivious State Preparation

Server | **Security requirement: Server gains no (or negligible) information about r** | **Client**

$|\psi\rangle$ →

← $Z^b P^r |\psi_1\rangle$

Oblivious phase correction

← r

b →

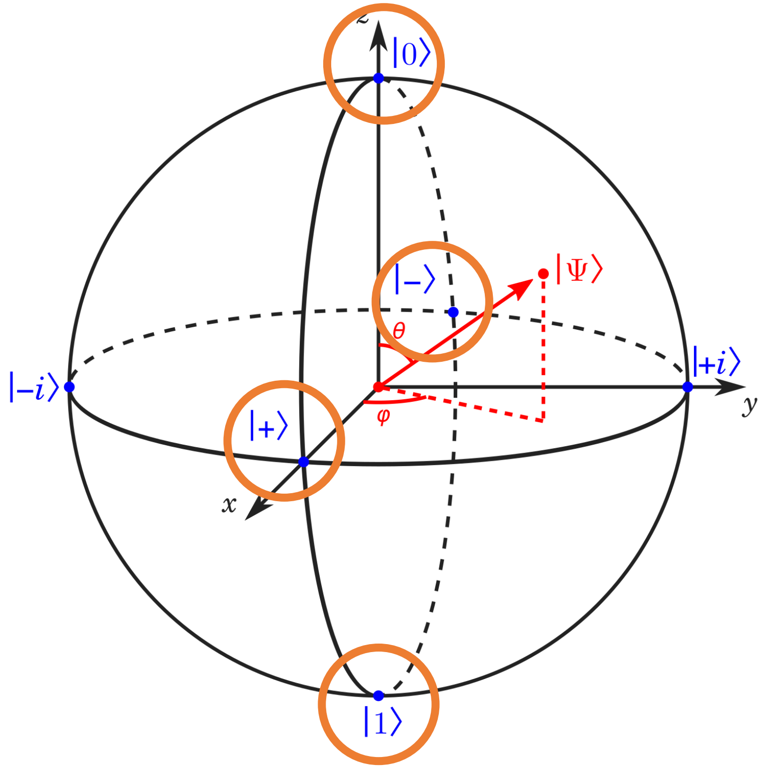
↓ Reduces to

← $Z^b P^r |+\rangle$

Oblivious state preparation

← r

b →



Easier task: Generate BB84 states, and then rotate

Oblivious Phase via Oblivious State Preparation

Server | **Client**

Security requirement: Server gains **no** (or negligible) information about r

Server → $|\psi\rangle$; ← $Z^b P^r |\psi_1\rangle$

Oblivious phase correction

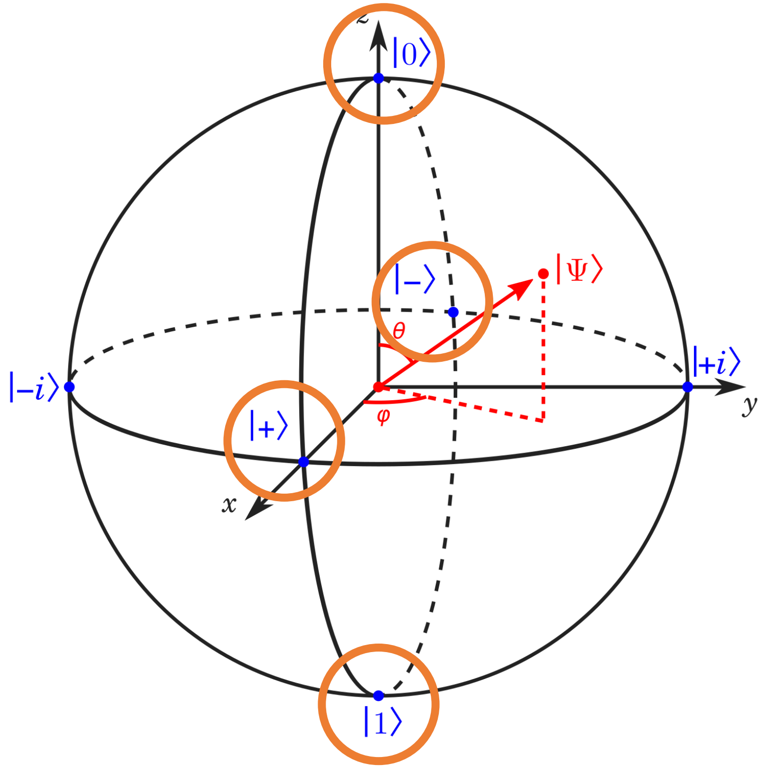
Client → r ; ← b

Reduces to

	$b = 0$	$b = 1$
$r = 0$:	$\|+\rangle$	$\|-\rangle$
$r = 1$:	$\|0\rangle$	$\|1\rangle$

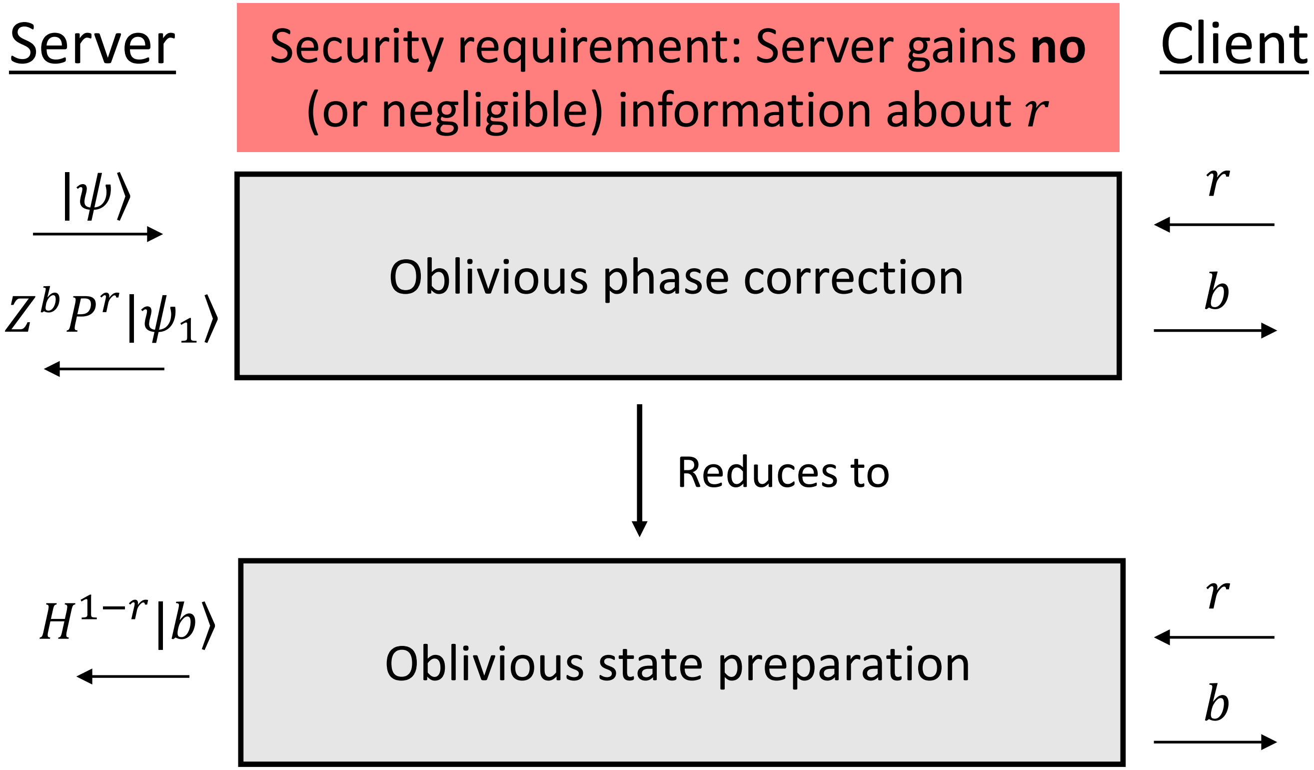
Oblivious state preparation

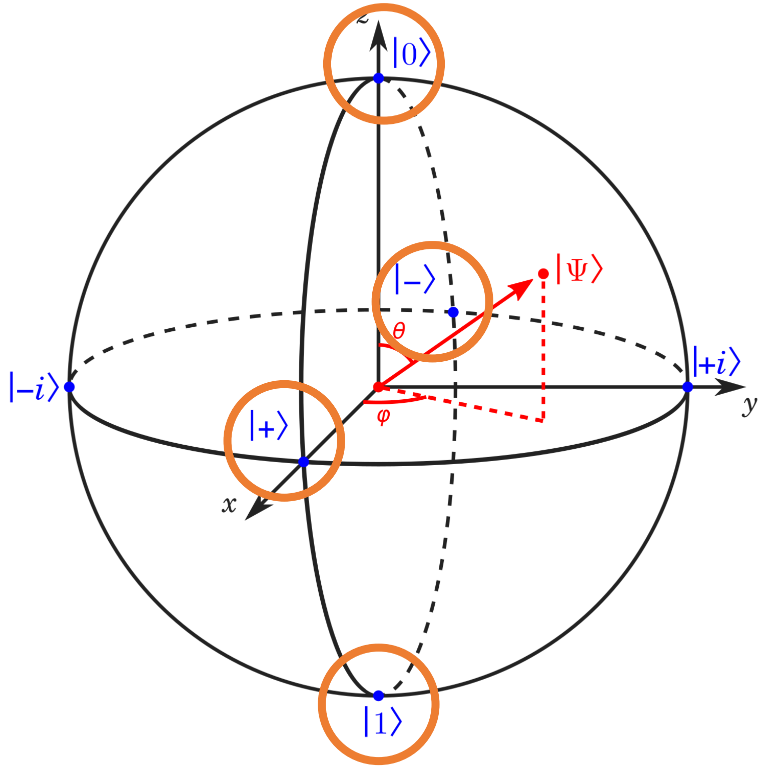
Client → r ; ← b



Easier task: Generate BB84 states, and then rotate

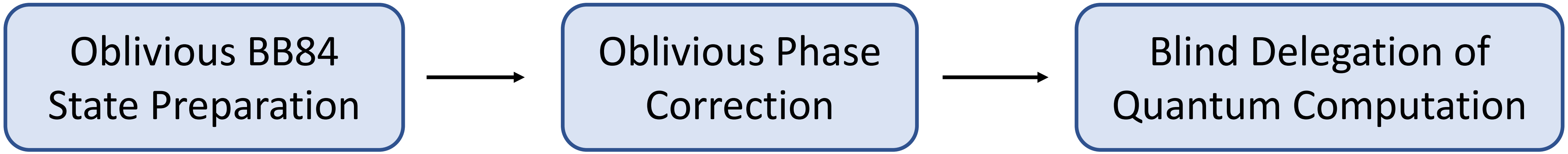
Oblivious Phase via Oblivious State Preparation





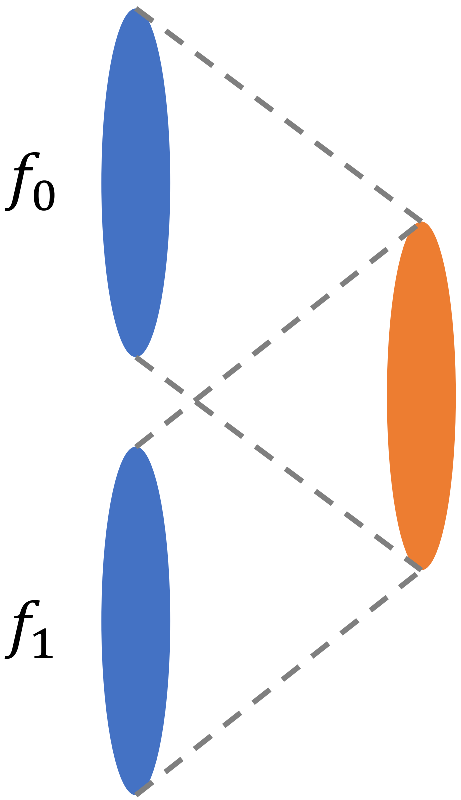
Easier task: Generate BB84 states, and then rotate

Progress so far…



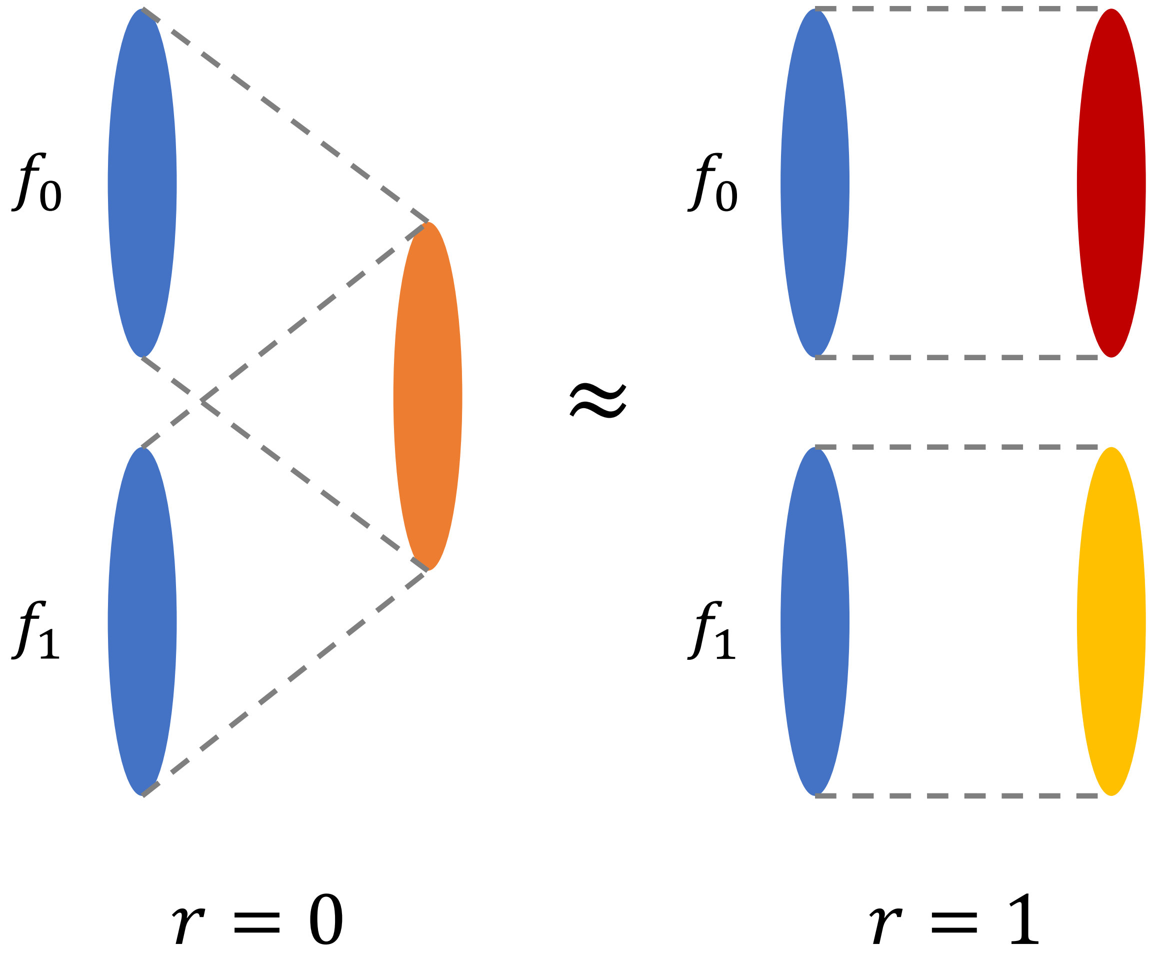
Part 3: How to Implement Oblivious BB84 State Preparation with Classical Communication

Key Tool: Trapdoor Claw-free Function (TCF)



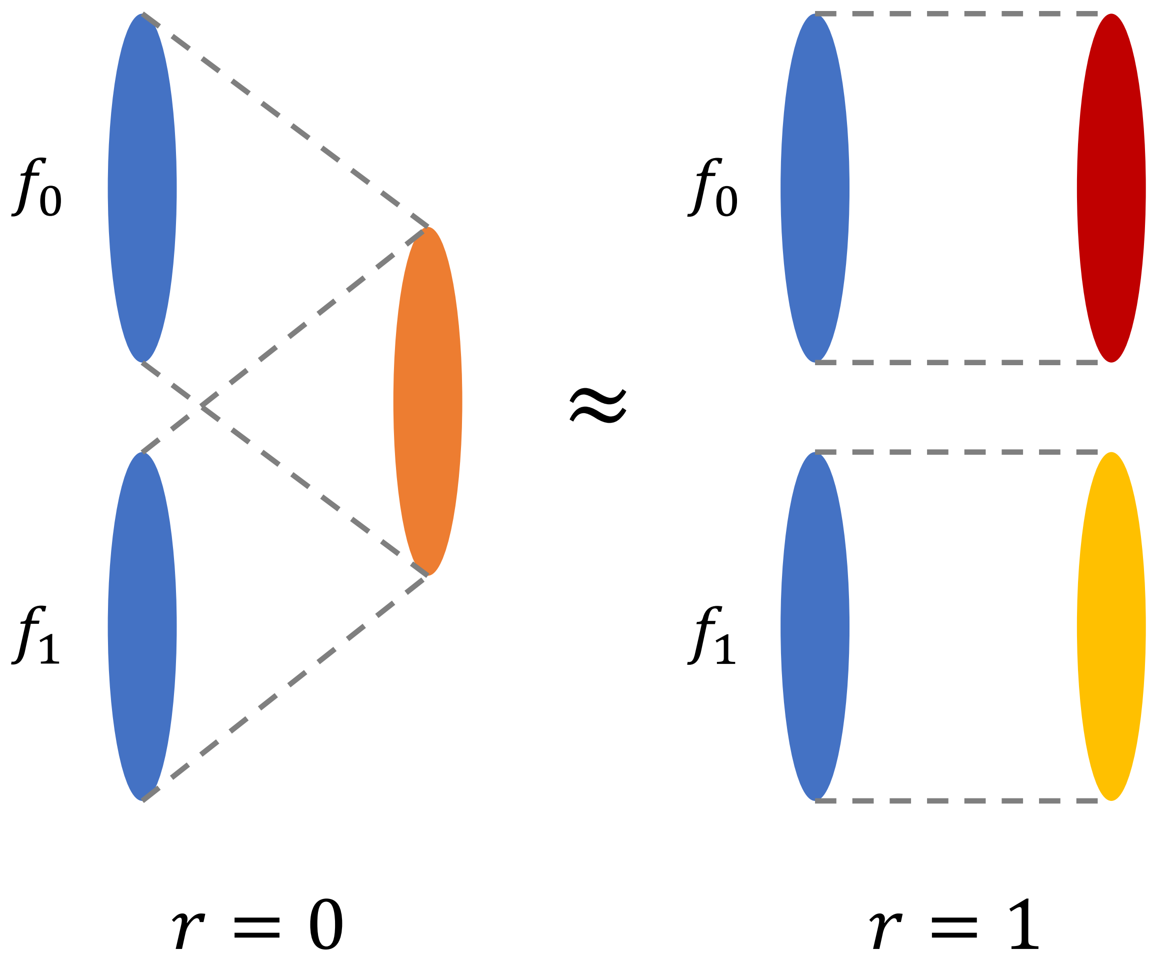
- Pair of injective functions $f_0, f_1: \mathcal{X} \to \mathcal{Y}$ such that for any $y \in \mathcal{Y}$, exists x_0, x_1 such that $f_0(x_0) = f_1(x_1) = y$
- Trapdoor: The Gen algorithm $(f_0, f_1, \mathrm{td}) \leftarrow \mathrm{Gen}$ outputs a trapdoor such that for any $y \in \mathcal{Y}$, $\mathrm{Invert}(\mathrm{td}, y) = x_0, x_1$
- Claw-free: Given f_0, f_1, no polynomial-time adversary can find a "claw" x_0, x_1 such that $f_0(x_0) = f_1(x_1)$

Dual-Mode Trapdoor Claw-free Function (dTCF)



- Pair of injective functions $f_0, f_1: \mathcal{X} \rightarrow \mathcal{Y}$ such that for any $y \in \mathcal{Y}$, exists x_0, x_1 such that $f_0(x_0) = f_1(x_1) = y$
- Trapdoor: The Gen algorithm $(f_0, f_1, \mathrm{td}) \leftarrow \mathrm{Gen}(r)$ outputs a trapdoor such that for any $y \in \mathcal{Y}$, $\mathrm{Invert}(\mathrm{td}, y) = x_0, x_1$
- Claw-free: Given f_0, f_1, no polynomial-time adversary can find a "claw" x_0, x_1 such that $f_0(x_0) = f_1(x_1)$

Dual-Mode Trapdoor Claw-free Function (dTCF)



- Pair of injective functions $f_0, f_1: \mathcal{X} \to \mathcal{Y}$ such that for any $y \in \mathcal{Y}$, exists x_0, x_1 such that $f_0(x_0) = f_1(x_1) = y$
- Trapdoor: The Gen algorithm $(f_0, f_1, \mathrm{td}) \leftarrow \mathrm{Gen}(r)$ outputs a trapdoor such that for any $y \in \mathcal{Y}$, $\mathrm{Invert}(\mathrm{td}, y) = x_0, x_1$
- Mode indistinguishability: $(f_0, f_1, \cdot) \leftarrow \mathrm{Gen}(0) \approx (f_0, f_1, \cdot) \leftarrow \mathrm{Gen}(1)$

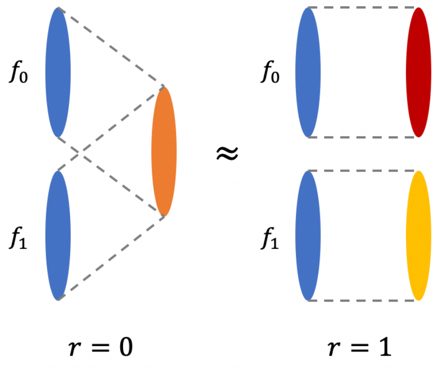
$H^{1-r}|b\rangle$ ←

Oblivious BB84 state preparation

← r

b →

Security requirement: Server gains **no** (or negligible) information about r



Server

Client

1. Prepare uniform superposition

$$\sum_{b\in\{0,1\},x\in\mathcal{X}}|b\rangle|x\rangle$$

2. Measure output of f_0, f_1

$$\sum_{b\in\{0,1\},x\in\mathcal{X}}|b\rangle|x\rangle|f_b(x)\rangle$$

↓ y

3. Measure input register in Hadamard basis

$\underline{r = 0}$

$$\sum_{b\in\{0,1\}}|b\rangle|x_b\rangle$$

$\underline{r = 1}$

$|b\rangle|x_b\rangle$

↓ ↓ d

$|b\rangle$

Sample $(f_0, f_1, \mathrm{td}) \leftarrow \mathrm{Gen}(r)$

← f_0, f_1

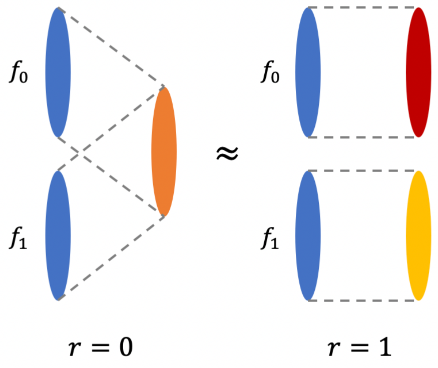
$H^{1-r}|b\rangle$ ←

Oblivious BB84 state preparation

← r

b →

Security requirement: Server gains **no** (or negligible) information about r



Server

Client

1. Prepare uniform superposition

$$\sum_{b\in\{0,1\},x\in\mathcal{X}}|b\rangle|x\rangle$$

2. Measure output of f_0, f_1

$$\sum_{b\in\{0,1\},x\in\mathcal{X}}|b\rangle|x\rangle|f_b(x)\rangle \rightarrow y$$

3. Measure input register in Hadamard basis

$r = 0$	$r = 1$
$\sum_{b\in\{0,1\}}\|b\rangle\|x_b\rangle$, $x_b \rightarrow d$	$\|b\rangle\|x_b\rangle$, $x_b \rightarrow d$
$Z^{d\cdot(x_0\oplus x_1)}(\|0\rangle + \|1\rangle)$	$\|b\rangle$

What happens when we measure the input register of a “claw state” in the Hadamard basis?

$$(I \otimes H^{\otimes\lambda})(\,|0\rangle|x_0\rangle + |1\rangle|x_1\rangle\,)$$

$$= |0\rangle \sum_{d\in\{0,1\}^\lambda}(-1)^{d\cdot x_0}|d\rangle + |1\rangle \sum_{d\in\{0,1\}^\lambda}(-1)^{d\cdot x_1}|d\rangle$$

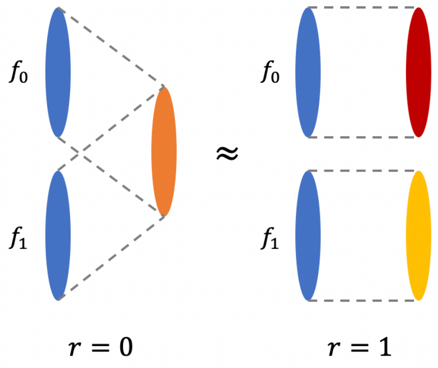
$$= \sum_{d\in\{0,1\}^\lambda}\left((-1)^{d\cdot x_0}|0\rangle + (-1)^{d\cdot x_1}|1\rangle\right)|d\rangle \quad (|d\rangle \rightarrow d)$$

$$\downarrow$$

$$(-1)^{d\cdot x_0}|0\rangle + (-1)^{d\cdot x_1}|1\rangle = |0\rangle + (-1)^{d\cdot(x_0\oplus x_1)}|1\rangle$$

$H^{1-r}|b\rangle$ ← **Oblivious BB84 state preparation** ← r, → b

Security requirement: Server gains **no** (or negligible) information about r



Server | **Client**

1. Prepare uniform superposition

$$\sum_{b\in\{0,1\},x\in\mathcal{X}}|b\rangle|x\rangle$$

Client: Sample $(f_0, f_1, \mathrm{td}) \leftarrow \mathrm{Gen}(r)$

2. Measure output of f_0, f_1 ← f_0, f_1

$$\sum_{b\in\{0,1\},x\in\mathcal{X}}|b\rangle|x\rangle|f_b(x)\rangle \to y$$

3. Measure input register in Hadamard basis

Server, $r = 0$: $\sum_{b\in\{0,1\}}|b\rangle|x_b\rangle \to d$ ⟹ $Z^{d\cdot(x_0\oplus x_1)}(|0\rangle + |1\rangle)$

Server, $r = 1$: $|b\rangle|x_b\rangle \to d$ ⟹ $|b\rangle$

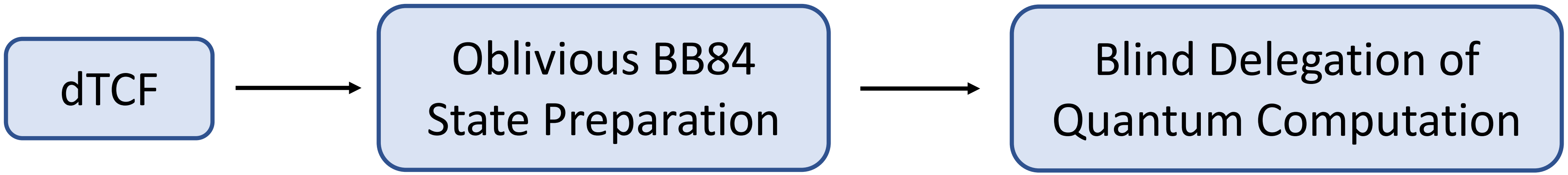
→ y, d →

Client, $r = 0$: $\mathrm{Invert}(\mathrm{td}, y) = x_0, x_1$ ⟹ $b = d\cdot(x_0 \oplus x_1)$

Client, $r = 1$: $\mathrm{Invert}(\mathrm{td}, y) = b, x_b$ ⟹ b

← output →

Progress so far…



dTCF from LWE

Basic idea:

Let q be a large modulus, $m > n$, and $A \in \mathbb{Z}_q^{m \times n}$ be a uniformly random matrix

Let $v = As$ for a uniformly random $s \in \mathbb{Z}_q^n$

Let
$$f_{(A,v),0}(x) = Ax$$
$$f_{(A,v),1}(x) = Ax + v = A(x + s)$$

On domain $x \in \mathbb{Z}_q^n$, this pair of functions have the same image

dTCF from LWE

Dual-mode:

Let q be a large modulus, $m > n$, and $A \in \mathbb{Z}_q^{m \times n}$ be a uniformly random matrix

Let $v = As$ for a uniformly random $s \in \mathbb{Z}_q^n$

Let
$$f_{(A,v),0}(x) = Ax$$
$$f_{(A,v),1}(x) = Ax + v$$

dTCF from LWE

Dual-mode:

Let q be a large modulus, $m > n$, and $A \in \mathbb{Z}_q^{m \times n}$ be a uniformly random matrix

If $r = 0$, sample $v \in \text{span}(A)$

If $r = 1$, sample $v \notin \text{span}(A)$

Let
$$f_{(A,v),0}(x) = Ax$$
$$f_{(A,v),1}(x) = Ax + v$$

dTCF from LWE

Dual-mode:

Let q be a large modulus, $m > n$, and $A \in \mathbb{Z}_q^{m \times n}$ be a uniformly random matrix

If $r = 0$, sample $v \in \text{span}(A)$

If $r = 1$, sample $v \leftarrow \mathbb{Z}_q^m$

Let $f_{(A,v),0}(x) = Ax$ and $f_{(A,v),1}(x) = Ax + v$

have the same image if $r = 0$

have disjoint images if $r = 1$

But… given (A, v), it is easy to distinguish whether $r = 0$ or $r = 1$

dTCF from LWE

Adding error:

Let q be a large modulus, $m > n$, and $A \in \mathbb{Z}_q^{m \times n}$ be a uniformly random matrix

If $r = 0$, sample (s, e), let $v = As + e$

If $r = 1$, sample $v \leftarrow \mathbb{Z}_q^m$

$e \in [-B, B]^m$, for $B \ll q$

Let
$$f_{(A,v),0}(x) = Ax$$
$$f_{(A,v),1}(x) = Ax + v$$

dTCF from LWE

Adding error:

Let q be a large modulus, $m > n$, and $A \in \mathbb{Z}_q^{m \times n}$ be a uniformly random matrix

If $r = 0$, sample (s, e), let $v = As + e$

If $r = 1$, sample $v \leftarrow \mathbb{Z}_q^m$

Let
$$f_{(A,v),0}(x) = Ax$$
$$f_{(A,v),1}(x) = Ax + v$$

Now, the $r = 0$ and $r = 1$ cases are indistinguishable assuming LWE!

New problem: when $r = 0$, functions no longer have the same image

dTCF from LWE

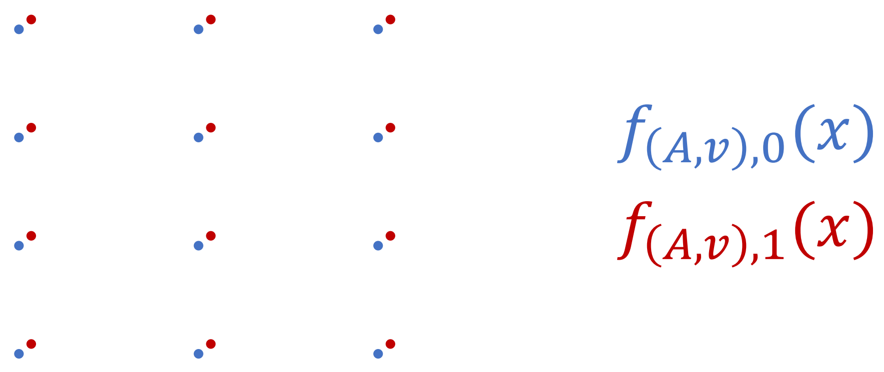
Adding error:

Let q be a large modulus, $m > n$, and $A \in \mathbb{Z}_q^{m \times n}$ be a uniformly random matrix

If $r = 0$, sample (s, e), let $v = As + e$

If $r = 1$, sample $v \leftarrow \mathbb{Z}_q^m$

Let
$$f_{(A,v),0}(x) = Ax$$
$$f_{(A,v),1}(x) = Ax + v$$
$$= A(x + s) + e$$



dTCF from LWE

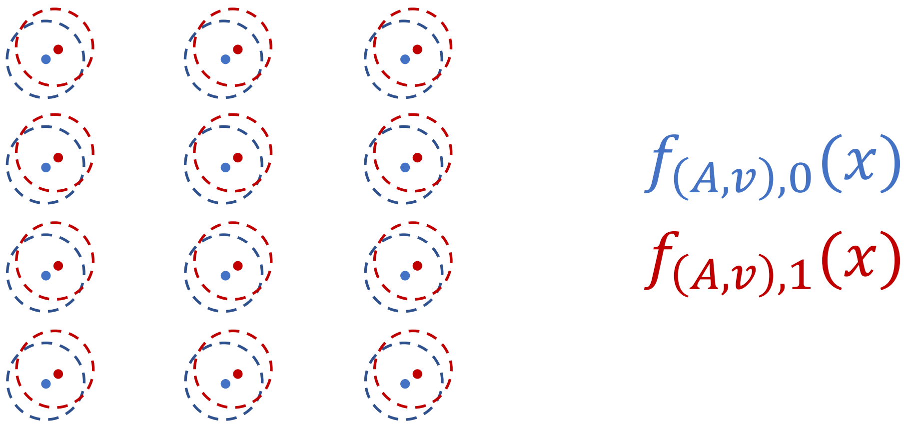
Adding error:

Let q be a large modulus, $m > n$, and $A \in \mathbb{Z}_q^{m \times n}$ be a uniformly random matrix

If $r = 0$, sample (s, e), let $v = As + e$ If $r = 1$, sample $v \leftarrow \mathbb{Z}_q^m$

Let
$$f_{(A,v),0}(x) = Ax$$
$$f_{(A,v),1}(x) = Ax + v$$
$$= A(x + s) + e$$

Solution:



dTCF from LWE

Adding error:

Let q be a large modulus, $m > n$, and $A \in \mathbb{Z}_q^{m \times n}$ be a uniformly random matrix

If $r = 0$, sample (s, e), let $v = As + e$

If $r = 1$, sample $v \leftarrow \mathbb{Z}_q^m$

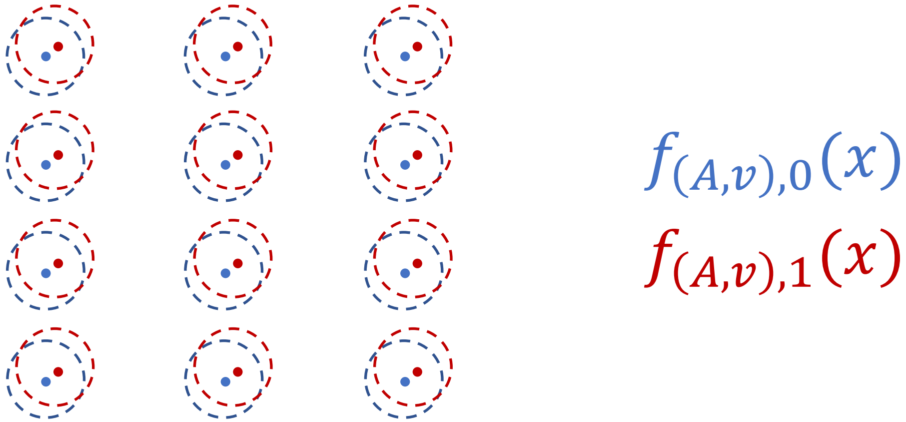
Let $f_{(A,v),0}(x) = Ax + e'$

$f_{(A,v),1}(x) = Ax + v + e'$

where $|e| \ll |e'| \ll q$

"noisy TCF"

Solution:



dTCF from LWE

Adding a trapdoor:

Let q be a large modulus, $m > n$, and $A \in \mathbb{Z}_q^{m \times n}$ be a uniformly random matrix

If $r = 0$, sample (s, e), let $v = As + e$

If $r = 1$, sample $v \leftarrow \mathbb{Z}_q^m$

Let
$$f_{(A,v),0}(x) = Ax + e'$$
$$f_{(A,v),1}(x) = Ax + v + e'$$

dTCF from LWE

Adding a trapdoor:

Over the reals

Sample $(A, T) \leftarrow \text{TrapGen}$: $A \in \mathbb{Z}_q^{m \times n}, TA = 0 \bmod q, T \in [-B, B]^{m \times m}$ is full rank

If $r = 0$, sample (s, e), let $v = As + e$

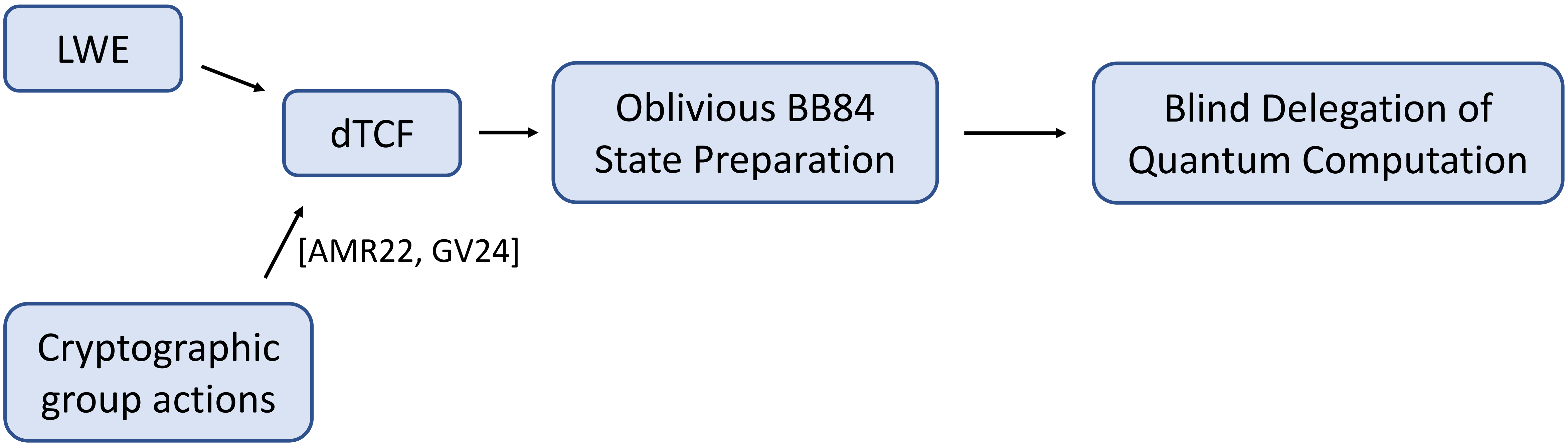
If $r = 1$, sample $v \leftarrow \mathbb{Z}_q^m$

Let
$$f_{(A,v),0}(x) = Ax + e'$$
$$f_{(A,v),1}(x) = Ax + v + e'$$

Let $\text{td} = T$

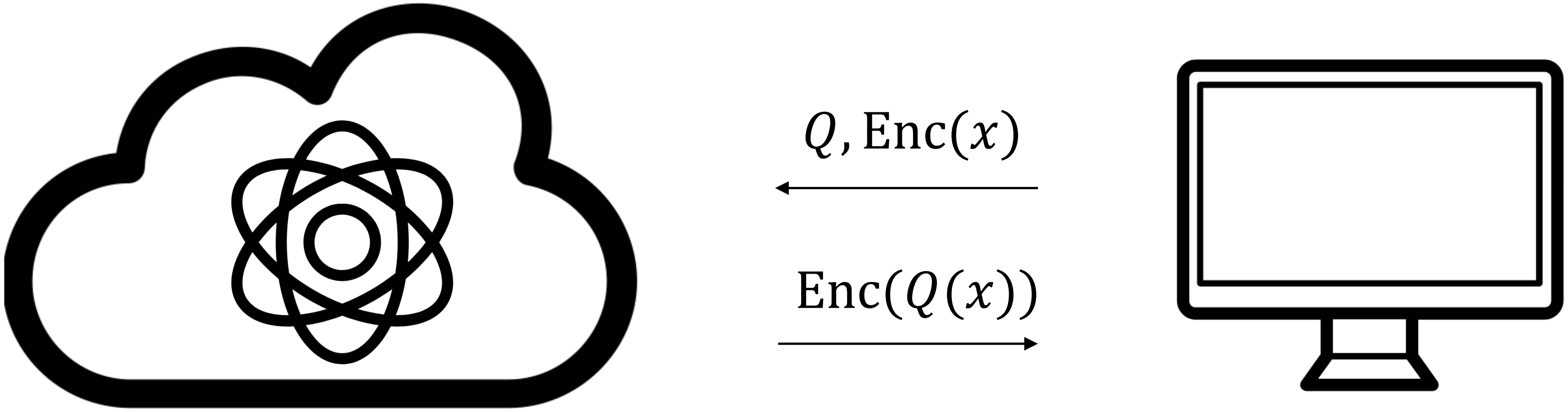
$\text{Invert}(\text{td}, Ax + e')$: Compute $T(Ax + e') \bmod q = Te'$, and solve for e'

Progress so far…



Quantum Fully-Homomorphic Encryption (QFHE)

- Minimally-interactive version of blind delegation



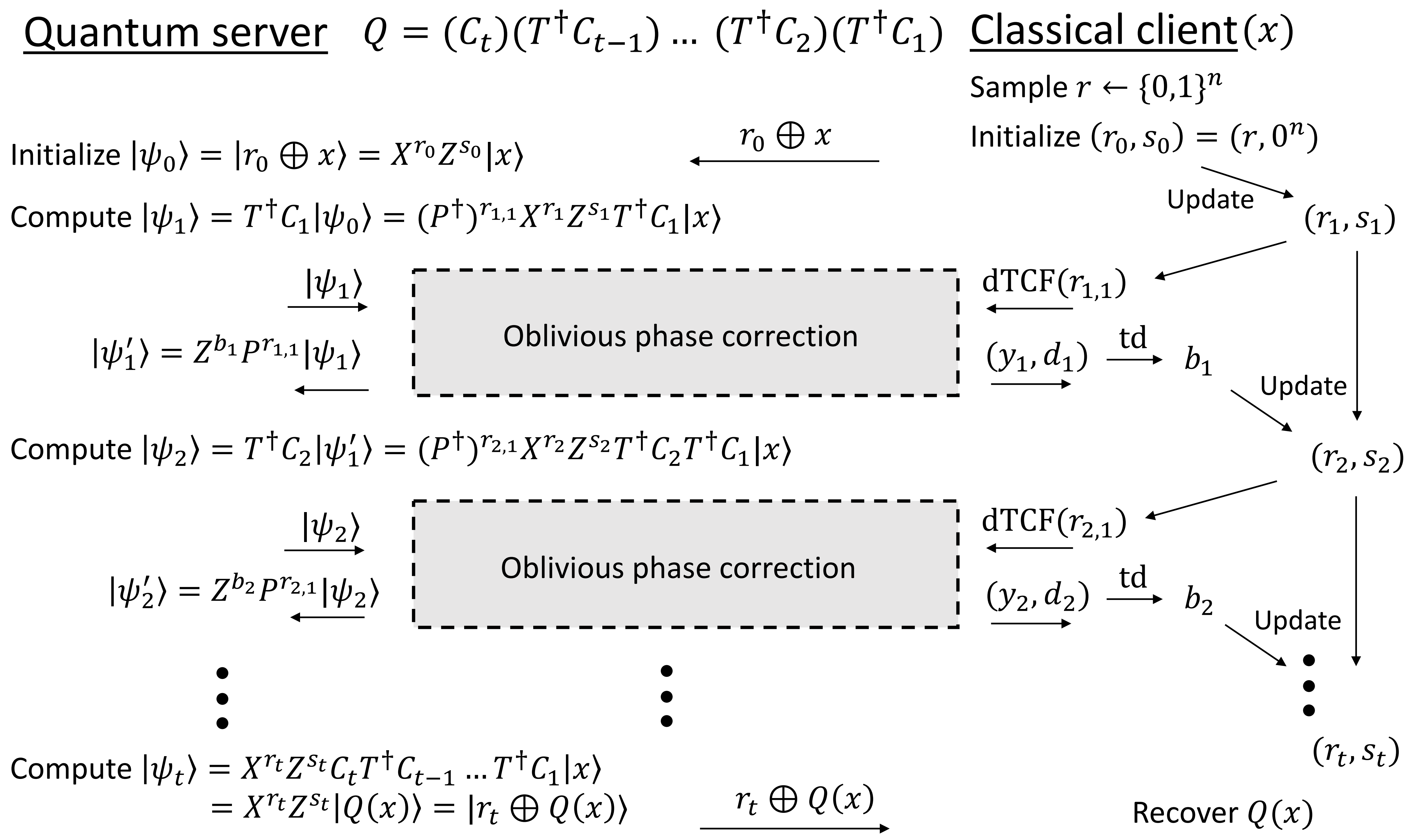
Quantum cloud

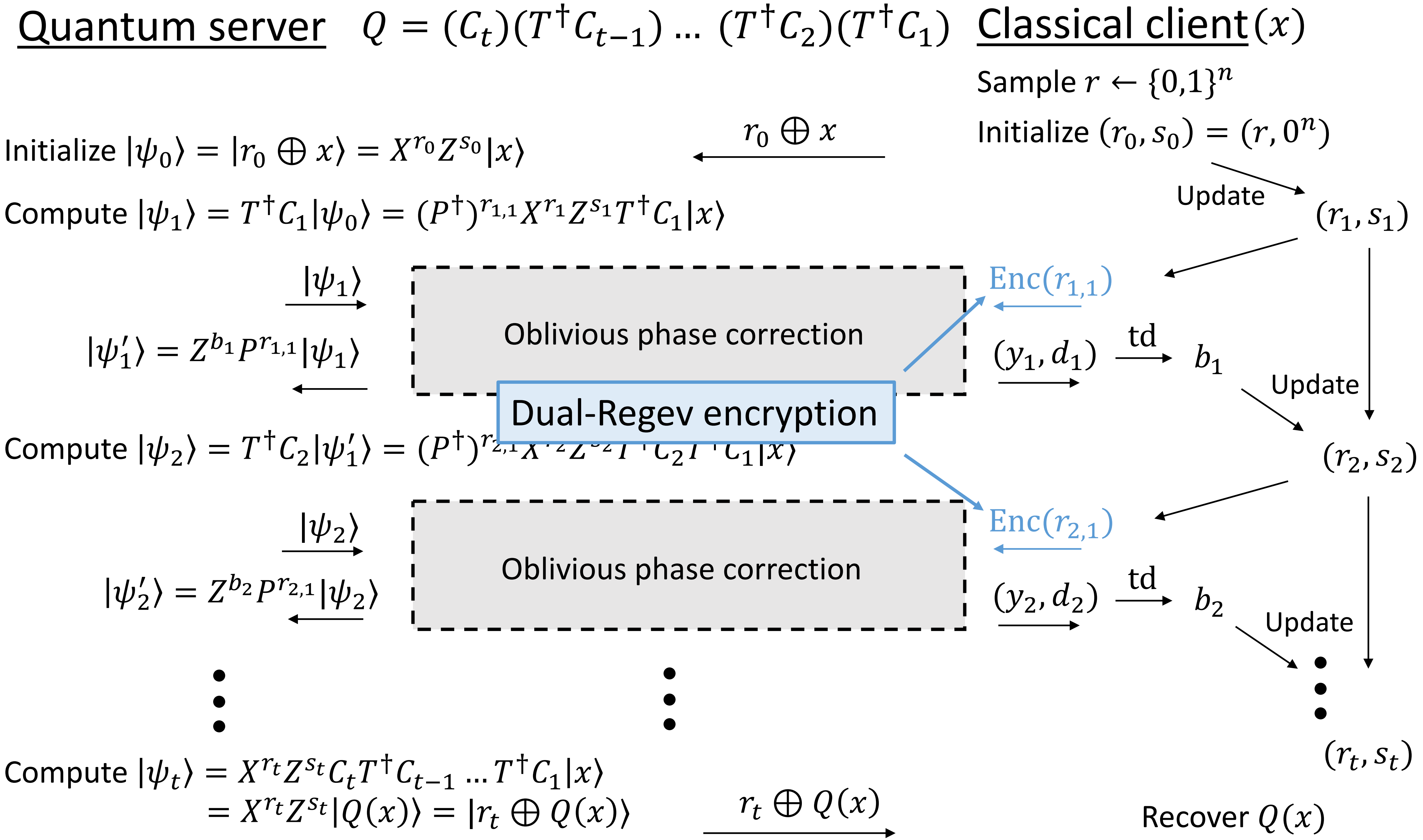
Classical client

- Observation [Mah17]: exists a classical FHE scheme such that $\mathrm{Enc}(r)$ is a dTCF with mode r

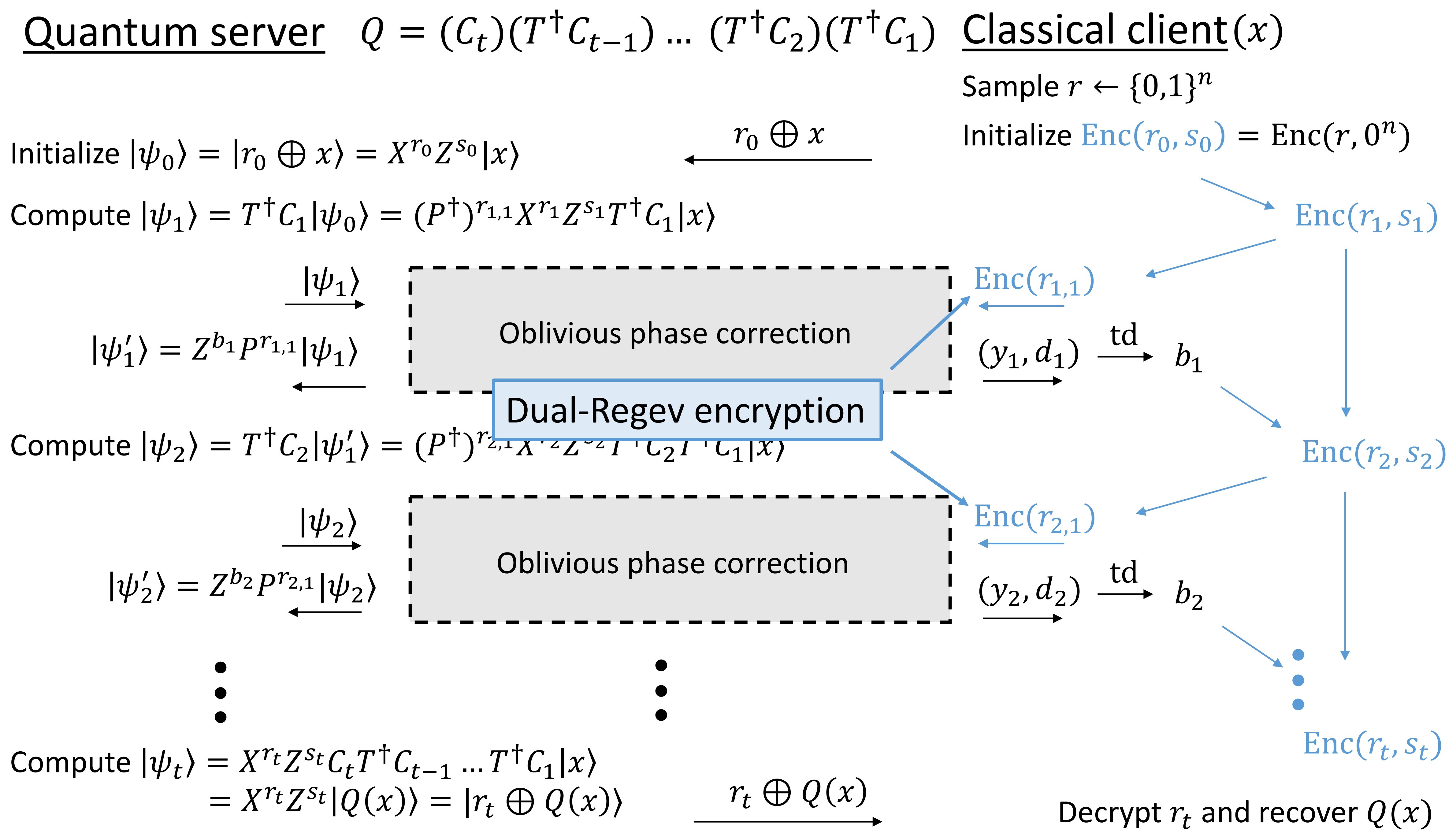
Dual-Regev Encryption

- KeyGen runs TrapGen to obtain $\mathrm{pk} = A, \mathrm{sk} = T$
- $\mathrm{Enc}(r \in \{0,1\}) \to As + e + r \cdot u$, where $u \notin \mathrm{span}(A)$ is a public vector
- This scheme can be extended to FHE (dual-GSW)
- Letting $v = \mathrm{Enc}(r)$, we have that (A, v) defines a dTCF with mode r

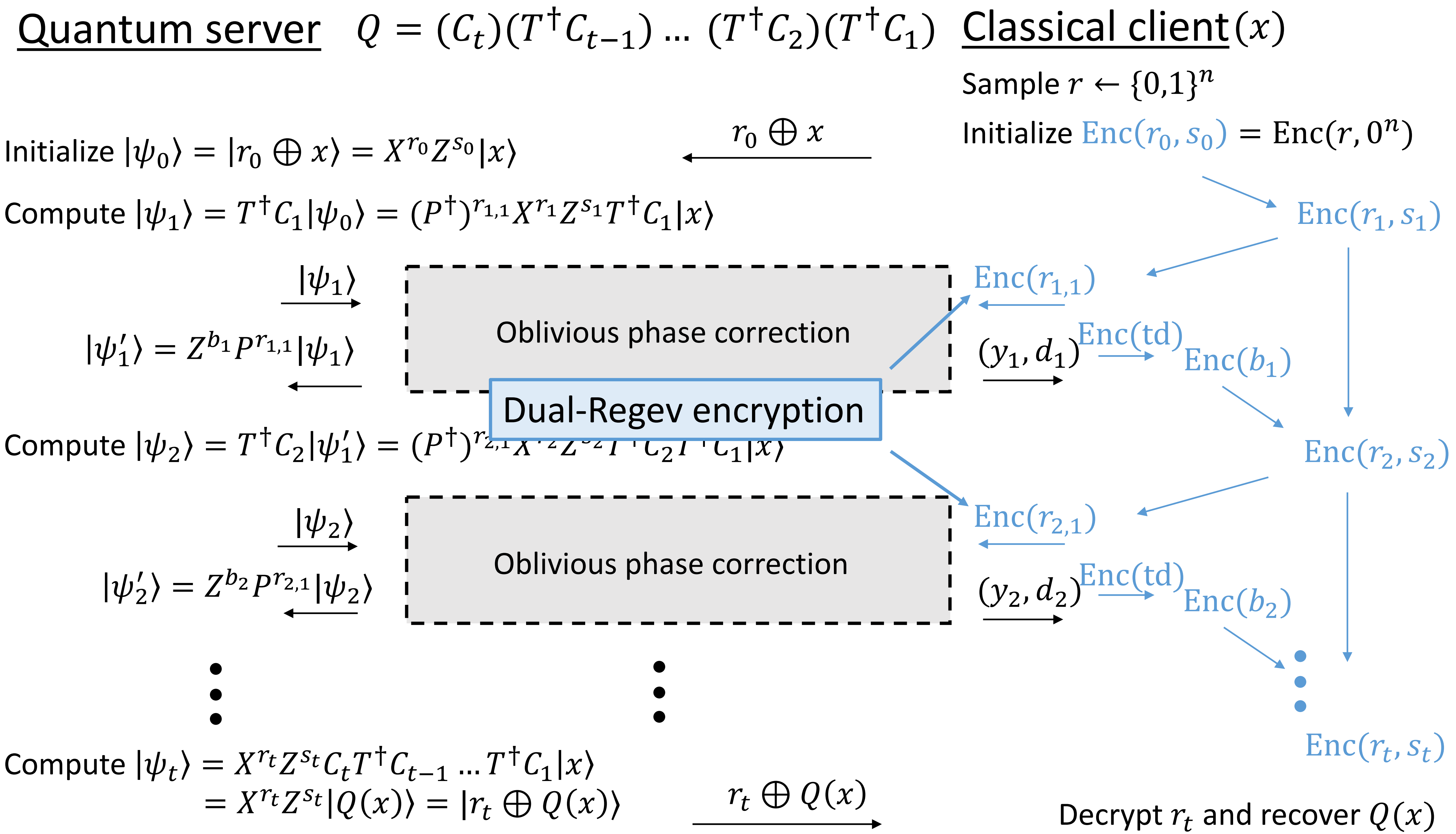

Quantum server
$Q = (C_t)(T^\dagger C_{t-1}) \dots (T^\dagger C_2)(T^\dagger C_1)$
Classical client(x)
Sample $r \leftarrow \{0,1\}^n$
Initialize $(r_0, s_0) = (r, 0^n)$
$r_0 \oplus x$
Initialize $|\psi_0\rangle = |r_0 \oplus x\rangle = X^{r_0} Z^{s_0} |x\rangle$
Update
(r_1, s_1)
Compute $|\psi_1\rangle = T^\dagger C_1 |\psi_0\rangle = (P^\dagger)^{r_{1,1}} X^{r_1} Z^{s_1} T^\dagger C_1 |x\rangle$
$|\psi_1\rangle$
Oblivious phase correction
$\mathrm{dTCF}(r_{1,1})$
$|\psi_1'\rangle = Z^{b_1} P^{r_{1,1}} |\psi_1\rangle$
(y_1, d_1)
td
b_1
Update
(r_2, s_2)
Compute $|\psi_2\rangle = T^\dagger C_2 |\psi_1'\rangle = (P^\dagger)^{r_{2,1}} X^{r_2} Z^{s_2} T^\dagger C_2 T^\dagger C_1 |x\rangle$
$|\psi_2\rangle$
Oblivious phase correction
$\mathrm{dTCF}(r_{2,1})$
$|\psi_2'\rangle = Z^{b_2} P^{r_{2,1}} |\psi_2\rangle$
(y_2, d_2)
td
b_2
Update
(r_t, s_t)
Compute $|\psi_t\rangle = X^{r_t} Z^{s_t} C_t T^\dagger C_{t-1} \dots T^\dagger C_1 |x\rangle$
$= X^{r_t} Z^{s_t} |Q(x)\rangle = |r_t \oplus Q(x)\rangle$
$r_t \oplus Q(x)$
Recover $Q(x)$

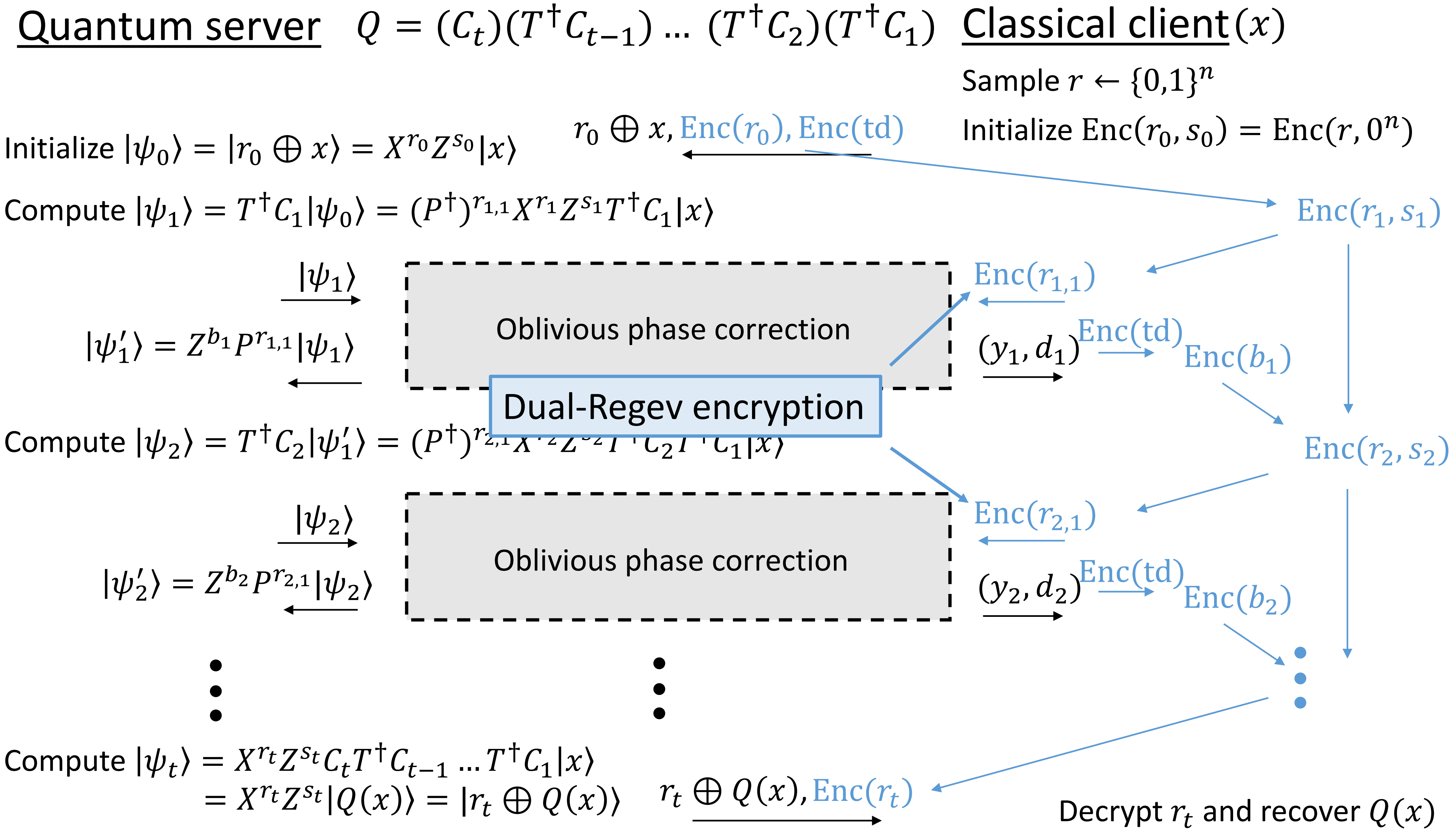


Quantum server
$Q = (C_t)(T^\dagger C_{t-1}) \dots (T^\dagger C_2)(T^\dagger C_1)$
Classical client (x)
Sample $r \leftarrow \{0,1\}^n$
Initialize $(r_0, s_0) = (r, 0^n)$
$r_0 \oplus x$
Initialize $|\psi_0\rangle = |r_0 \oplus x\rangle = X^{r_0} Z^{s_0} |x\rangle$
Update
(r_1, s_1)
Compute $|\psi_1\rangle = T^\dagger C_1 |\psi_0\rangle = (P^\dagger)^{r_{1,1}} X^{r_1} Z^{s_1} T^\dagger C_1 |x\rangle$
$|\psi_1\rangle$
Oblivious phase correction
$\mathrm{Enc}(r_{1,1})$
(y_1, d_1)
td
b_1
$|\psi_1'\rangle = Z^{b_1} P^{r_{1,1}} |\psi_1\rangle$
Update
Dual-Regev encryption
Compute $|\psi_2\rangle = T^\dagger C_2 |\psi_1'\rangle = (P^\dagger)^{r_{2,1}} X^{r_2} Z^{s_2} T^\dagger C_2 T^\dagger C_1 |x\rangle$
(r_2, s_2)
$|\psi_2\rangle$
Oblivious phase correction
$\mathrm{Enc}(r_{2,1})$
(y_2, d_2)
td
b_2
$|\psi_2'\rangle = Z^{b_2} P^{r_{2,1}} |\psi_2\rangle$
Update
(r_t, s_t)
Compute $|\psi_t\rangle = X^{r_t} Z^{s_t} C_t T^\dagger C_{t-1} \dots T^\dagger C_1 |x\rangle$
$= X^{r_t} Z^{s_t} |Q(x)\rangle = |r_t \oplus Q(x)\rangle$
$r_t \oplus Q(x)$
Recover $Q(x)$

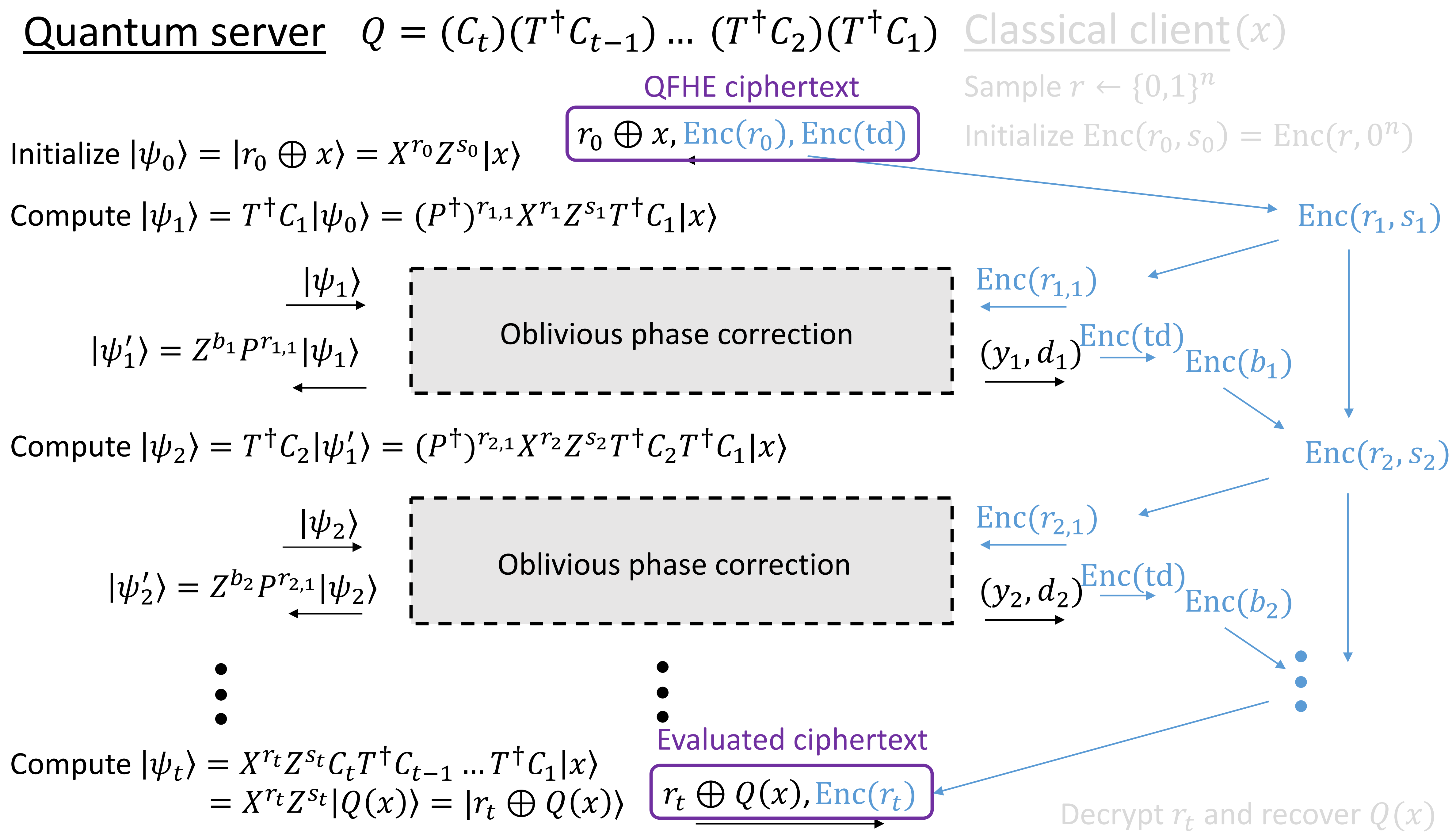


Quantum server
$Q = (C_t)(T^\dagger C_{t-1}) \dots (T^\dagger C_2)(T^\dagger C_1)$
Classical client (x)
Sample $r \leftarrow \{0,1\}^n$
Initialize $\mathrm{Enc}(r_0, s_0) = \mathrm{Enc}(r, 0^n)$
$r_0 \oplus x$
Initialize $|\psi_0\rangle = |r_0 \oplus x\rangle = X^{r_0} Z^{s_0} |x\rangle$
Compute $|\psi_1\rangle = T^\dagger C_1 |\psi_0\rangle = (P^\dagger)^{r_{1,1}} X^{r_1} Z^{s_1} T^\dagger C_1 |x\rangle$
$\mathrm{Enc}(r_1, s_1)$
$|\psi_1\rangle$
Oblivious phase correction
$\mathrm{Enc}(r_{1,1})$
$|\psi_1'\rangle = Z^{b_1} P^{r_{1,1}} |\psi_1\rangle$
(y_1, d_1)
td
b_1
Dual-Regev encryption
Compute $|\psi_2\rangle = T^\dagger C_2 |\psi_1'\rangle = (P^\dagger)^{r_{2,1}} X^{r_2} Z^{s_2} T^\dagger C_2 T^\dagger C_1 |x\rangle$
$\mathrm{Enc}(r_2, s_2)$
$|\psi_2\rangle$
Oblivious phase correction
$\mathrm{Enc}(r_{2,1})$
$|\psi_2'\rangle = Z^{b_2} P^{r_{2,1}} |\psi_2\rangle$
(y_2, d_2)
td
b_2
$\mathrm{Enc}(r_t, s_t)$
Compute $|\psi_t\rangle = X^{r_t} Z^{s_t} C_t T^\dagger C_{t-1} \dots T^\dagger C_1 |x\rangle$
$= X^{r_t} Z^{s_t} |Q(x)\rangle = |r_t \oplus Q(x)\rangle$
$r_t \oplus Q(x)$
Decrypt r_t and recover $Q(x)$


Quantum server
$Q = (C_t)(T^\dagger C_{t-1}) \dots (T^\dagger C_2)(T^\dagger C_1)$
Classical client (x)
Sample $r \leftarrow \{0,1\}^n$
Initialize $\mathrm{Enc}(r_0, s_0) = \mathrm{Enc}(r, 0^n)$
$r_0 \oplus x$
Initialize $|\psi_0\rangle = |r_0 \oplus x\rangle = X^{r_0} Z^{s_0} |x\rangle$
Compute $|\psi_1\rangle = T^\dagger C_1 |\psi_0\rangle = (P^\dagger)^{r_{1,1}} X^{r_1} Z^{s_1} T^\dagger C_1 |x\rangle$
$\mathrm{Enc}(r_1, s_1)$
$|\psi_1\rangle$
Oblivious phase correction
$\mathrm{Enc}(r_{1,1})$
$|\psi_1'\rangle = Z^{b_1} P^{r_{1,1}} |\psi_1\rangle$
(y_1, d_1)
$\mathrm{Enc}(\mathrm{td})$
$\mathrm{Enc}(b_1)$
Dual-Regev encryption
Compute $|\psi_2\rangle = T^\dagger C_2 |\psi_1'\rangle = (P^\dagger)^{r_{2,1}} X^{r_2} Z^{s_2} T^\dagger C_2 T^\dagger C_1 |x\rangle$
$\mathrm{Enc}(r_2, s_2)$
$|\psi_2\rangle$
Oblivious phase correction
$\mathrm{Enc}(r_{2,1})$
$|\psi_2'\rangle = Z^{b_2} P^{r_{2,1}} |\psi_2\rangle$
(y_2, d_2)
$\mathrm{Enc}(\mathrm{td})$
$\mathrm{Enc}(b_2)$
$\mathrm{Enc}(r_t, s_t)$
Compute $|\psi_t\rangle = X^{r_t} Z^{s_t} C_t T^\dagger C_{t-1} \dots T^\dagger C_1 |x\rangle$
$= X^{r_t} Z^{s_t} |Q(x)\rangle = |r_t \oplus Q(x)\rangle$
$r_t \oplus Q(x)$
Decrypt r_t and recover $Q(x)$



Quantum server
$Q = (C_t)(T^\dagger C_{t-1}) \dots (T^\dagger C_2)(T^\dagger C_1)$
Classical client (x)
Sample $r \leftarrow \{0,1\}^n$
$r_0 \oplus x$, $\mathrm{Enc}(r_0)$, $\mathrm{Enc}(\mathrm{td})$
Initialize $\mathrm{Enc}(r_0, s_0) = \mathrm{Enc}(r, 0^n)$
Initialize $|\psi_0\rangle = |r_0 \oplus x\rangle = X^{r_0} Z^{s_0} |x\rangle$
Compute $|\psi_1\rangle = T^\dagger C_1 |\psi_0\rangle = (P^\dagger)^{r_{1,1}} X^{r_1} Z^{s_1} T^\dagger C_1 |x\rangle$
$\mathrm{Enc}(r_1, s_1)$
$|\psi_1\rangle$
$\mathrm{Enc}(r_{1,1})$
Oblivious phase correction
$|\psi_1'\rangle = Z^{b_1} P^{r_{1,1}} |\psi_1\rangle$
(y_1, d_1)
$\mathrm{Enc}(\mathrm{td})$
$\mathrm{Enc}(b_1)$
Dual-Regev encryption
Compute $|\psi_2\rangle = T^\dagger C_2 |\psi_1'\rangle = (P^\dagger)^{r_{2,1}} X^{r_2} Z^{s_2} T^\dagger C_2 T^\dagger C_1 |x\rangle$
$\mathrm{Enc}(r_2, s_2)$
$|\psi_2\rangle$
$\mathrm{Enc}(r_{2,1})$
Oblivious phase correction
$|\psi_2'\rangle = Z^{b_2} P^{r_{2,1}} |\psi_2\rangle$
(y_2, d_2)
$\mathrm{Enc}(\mathrm{td})$
$\mathrm{Enc}(b_2)$
Compute $|\psi_t\rangle = X^{r_t} Z^{s_t} C_t T^\dagger C_{t-1} \dots T^\dagger C_1 |x\rangle$
$= X^{r_t} Z^{s_t} |Q(x)\rangle = |r_t \oplus Q(x)\rangle$
$r_t \oplus Q(x)$, $\mathrm{Enc}(r_t)$
Decrypt r_t and recover $Q(x)$


Quantum server
$Q = (C_t)(T^\dagger C_{t-1}) \ldots (T^\dagger C_2)(T^\dagger C_1)$
Classical client (x)
QFHE ciphertext
$r_0 \oplus x$, $\mathrm{Enc}(r_0)$, $\mathrm{Enc}(\mathrm{td})$
Sample $r \leftarrow \{0,1\}^n$
Initialize $\mathrm{Enc}(r_0, s_0) = \mathrm{Enc}(r, 0^n)$
Initialize $|\psi_0\rangle = |r_0 \oplus x\rangle = X^{r_0} Z^{s_0} |x\rangle$
Compute $|\psi_1\rangle = T^\dagger C_1 |\psi_0\rangle = (P^\dagger)^{r_{1,1}} X^{r_1} Z^{s_1} T^\dagger C_1 |x\rangle$
$\mathrm{Enc}(r_1, s_1)$
$|\psi_1\rangle$
Oblivious phase correction
$\mathrm{Enc}(r_{1,1})$
$|\psi_1'\rangle = Z^{b_1} P^{r_{1,1}} |\psi_1\rangle$
(y_1, d_1)
$\mathrm{Enc}(\mathrm{td})$
$\mathrm{Enc}(b_1)$
Compute $|\psi_2\rangle = T^\dagger C_2 |\psi_1'\rangle = (P^\dagger)^{r_{2,1}} X^{r_2} Z^{s_2} T^\dagger C_2 T^\dagger C_1 |x\rangle$
$\mathrm{Enc}(r_2, s_2)$
$|\psi_2\rangle$
Oblivious phase correction
$\mathrm{Enc}(r_{2,1})$
$|\psi_2'\rangle = Z^{b_2} P^{r_{2,1}} |\psi_2\rangle$
(y_2, d_2)
$\mathrm{Enc}(\mathrm{td})$
$\mathrm{Enc}(b_2)$
Evaluated ciphertext
Compute $|\psi_t\rangle = X^{r_t} Z^{s_t} C_t T^\dagger C_{t-1} \ldots T^\dagger C_1 |x\rangle$
$= X^{r_t} Z^{s_t} |Q(x)\rangle = |r_t \oplus Q(x)\rangle$
$r_t \oplus Q(x)$, $\mathrm{Enc}(r_t)$
Decrypt r_t and recover $Q(x)$

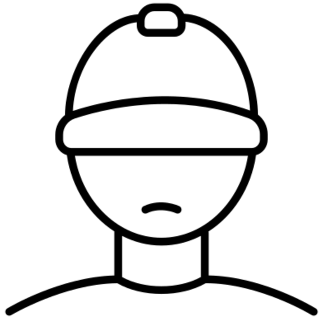
Part 4: Proofs of Quantumness and Verifiable Delegation

CHSH (Clauser, Horne, Shimony, Holt) Game

Alice

No communication

Bob



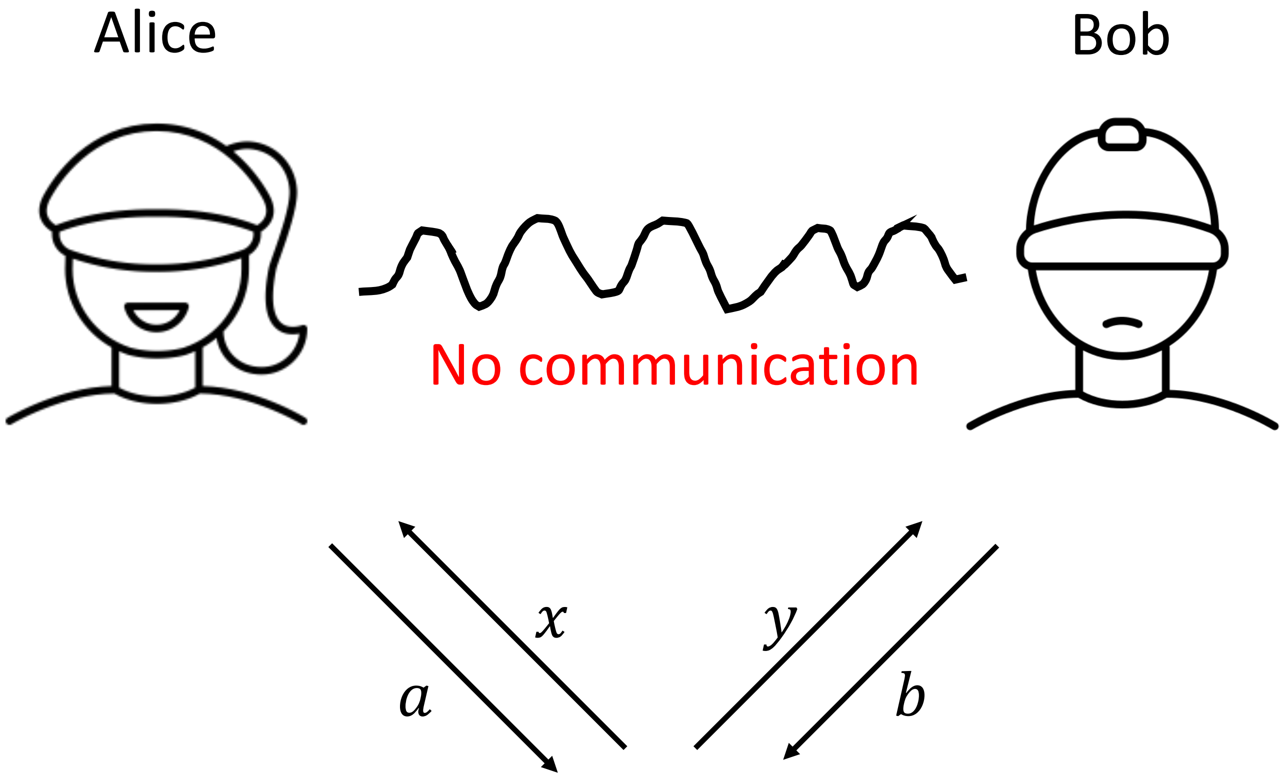
Because Bob has no information about Alice's question, any strategy is stuck at ¾

- One strategy: always set $a = b = 0$
- Wins with probability ¾
- Is this optimal?
- Yes: "Classical value" of CHSH is $\omega_{\mathrm{CHSH}} = \frac{3}{4}$

Fix any deterministic Alice strategy $f_A(x)$

Winning condition	Case 1: $f_A(0) = f_A(1)$	Case 2: $f_A(0) \neq f_A(1)$
$f_B(0) = f_A(x)$	1	½
$f_B(1) = x \oplus f_A(x)$	½	1
Win probability:	¾	¾

CHSH with quantum entangled strategies



Sample $x, y \leftarrow \{0,1\}$

They win if $a \oplus b = x \wedge y$

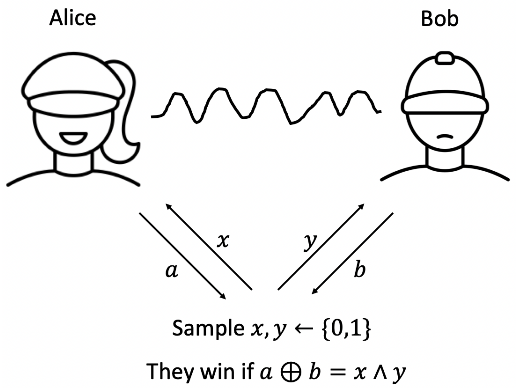
Can they do better than ¾?

What is the “quantum value” ω^*_{CHSH} of CHSH?

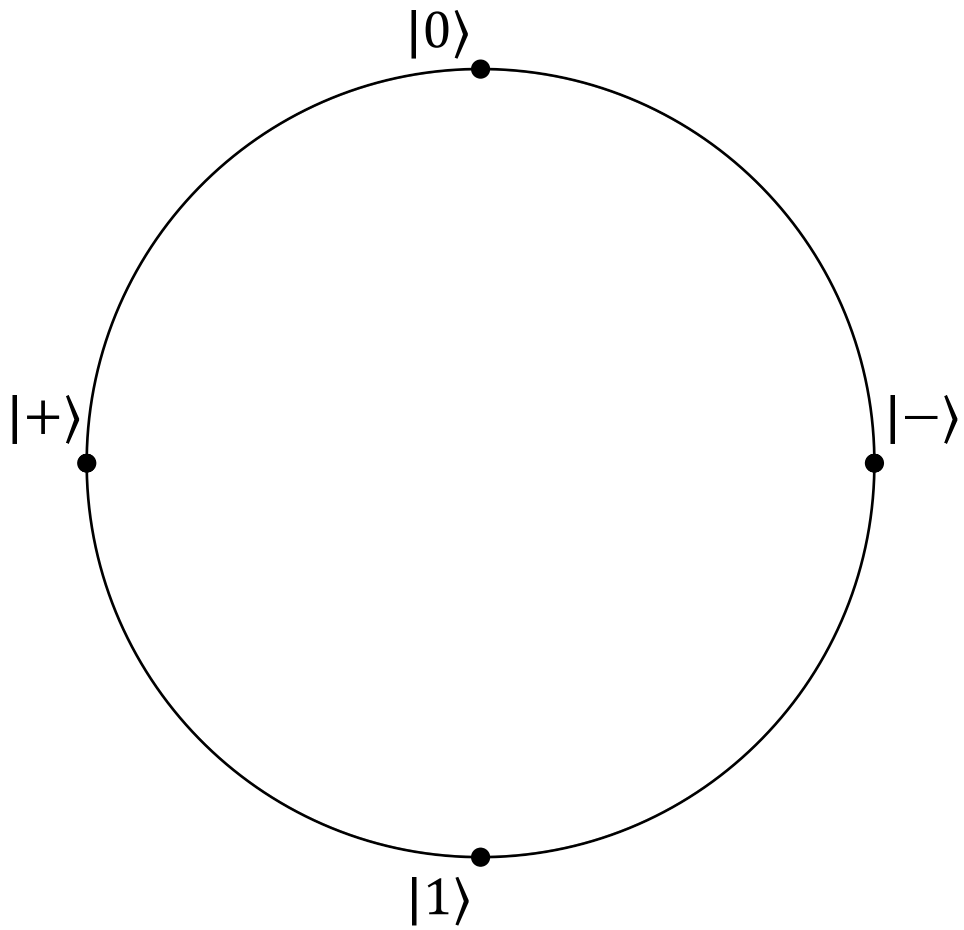
CHSH with quantum entangled strategies

Start with an EPR pair:

$|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B = |+\rangle_A|+\rangle_B + |-\rangle_A|-\rangle_B$



Bob's view:

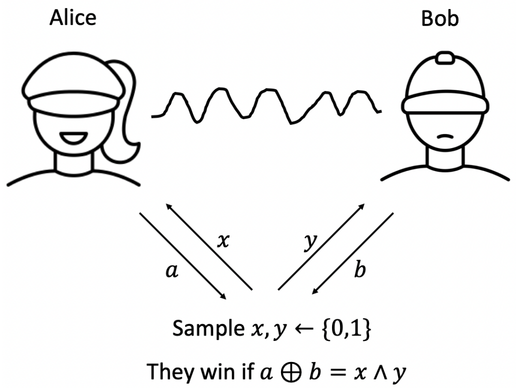


Alice: if $x = 0$, measure in the Hadamard basis (X)
if $x = 1$, measure in the standard basis (Z)
let a be the bit measured

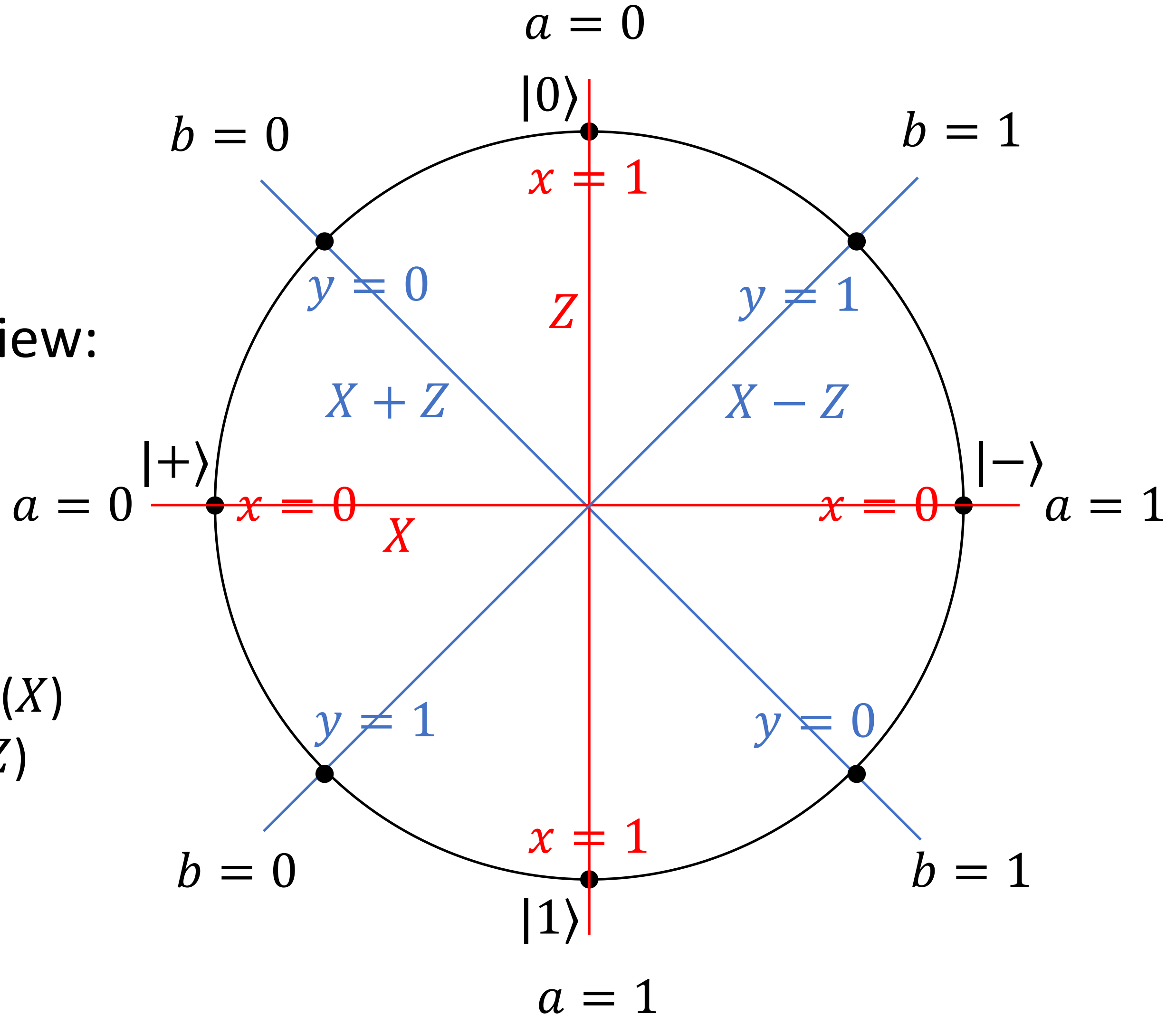
CHSH with quantum entangled strategies

Start with an EPR pair:

$$|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B = |+\rangle_A|+\rangle_B + |-\rangle_A|-\rangle_B$$



Bob's view:



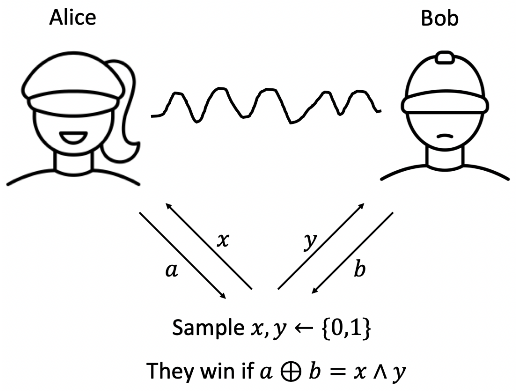
Alice: if $x = 0$, measure in the Hadamard basis (X)
if $x = 1$, measure in the standard basis (Z)
let a be the bit measured

Bob: if $y = 0$, measure in the $X + Z$ basis
if $y = 1$, measure in the $X - Z$ basis
let b be the bit measured

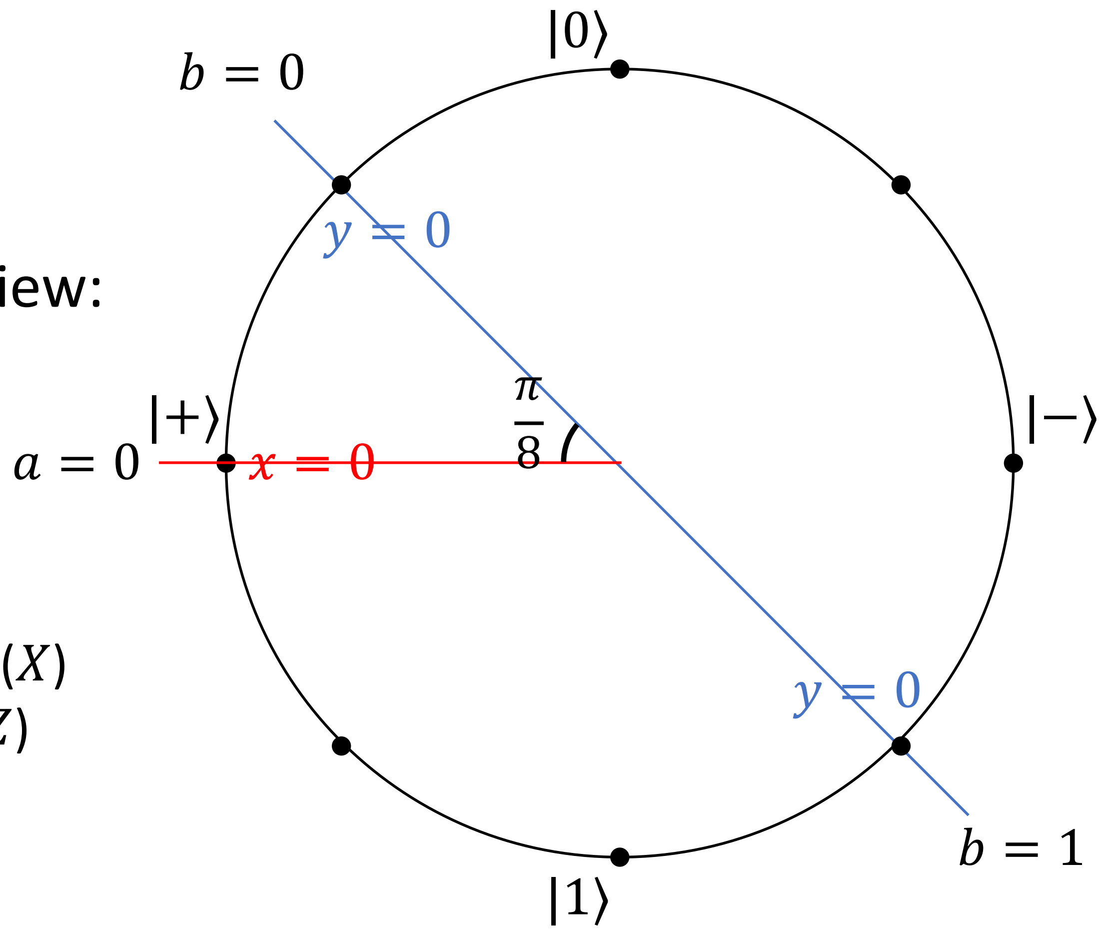
CHSH with quantum entangled strategies

Start with an EPR pair:

$$|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B = |+\rangle_A|+\rangle_B + |-\rangle_A|-\rangle_B$$



Bob's view:



Alice: if $x = 0$, measure in the Hadamard basis (X)
if $x = 1$, measure in the standard basis (Z)
let a be the bit measured

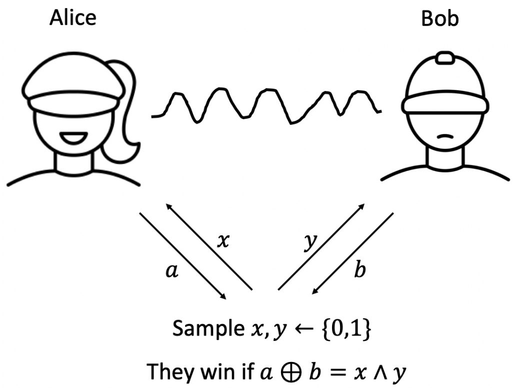
Bob: if $y = 0$, measure in the $X + Z$ basis
if $y = 1$, measure in the $X - Z$ basis
let b be the bit measured

Example: $x = 0, a = 0, y = 0 \rightarrow$ win when $b = 0 \rightarrow$ Pr $\cos^2(\frac{\pi}{8}) \approx 0.85$

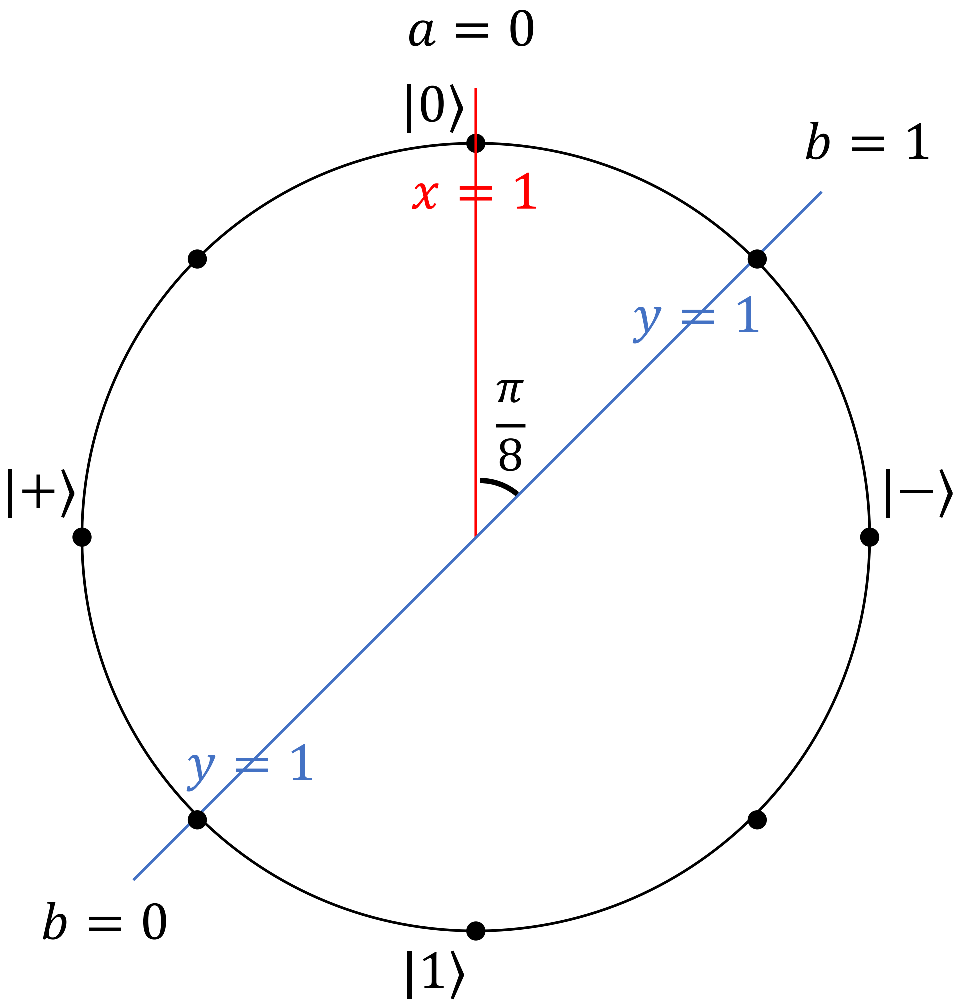
CHSH with quantum entangled strategies

Start with an EPR pair:

$|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B = |+\rangle_A|+\rangle_B + |-\rangle_A|-\rangle_B$



Bob's view:



Alice: if $x = 0$, measure in the Hadamard basis (X)
if $x = 1$, measure in the standard basis (Z)
let a be the bit measured

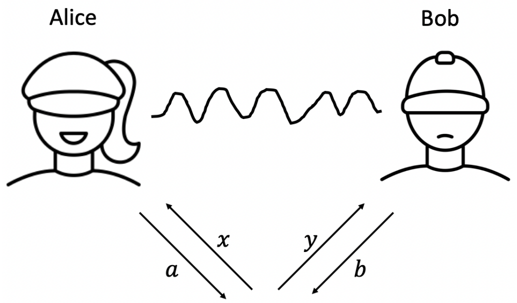
Bob: if $y = 0$, measure in the $X + Z$ basis
if $y = 1$, measure in the $X - Z$ basis
let b be the bit measured

Example: $x = 1, a = 0, y = 1 \;\rightarrow\;$ win when $b = 1 \rightarrow$ Pr $\cos^2(\frac{\pi}{8}) \approx 0.85$

CHSH with quantum entangled strategies

Start with an EPR pair:

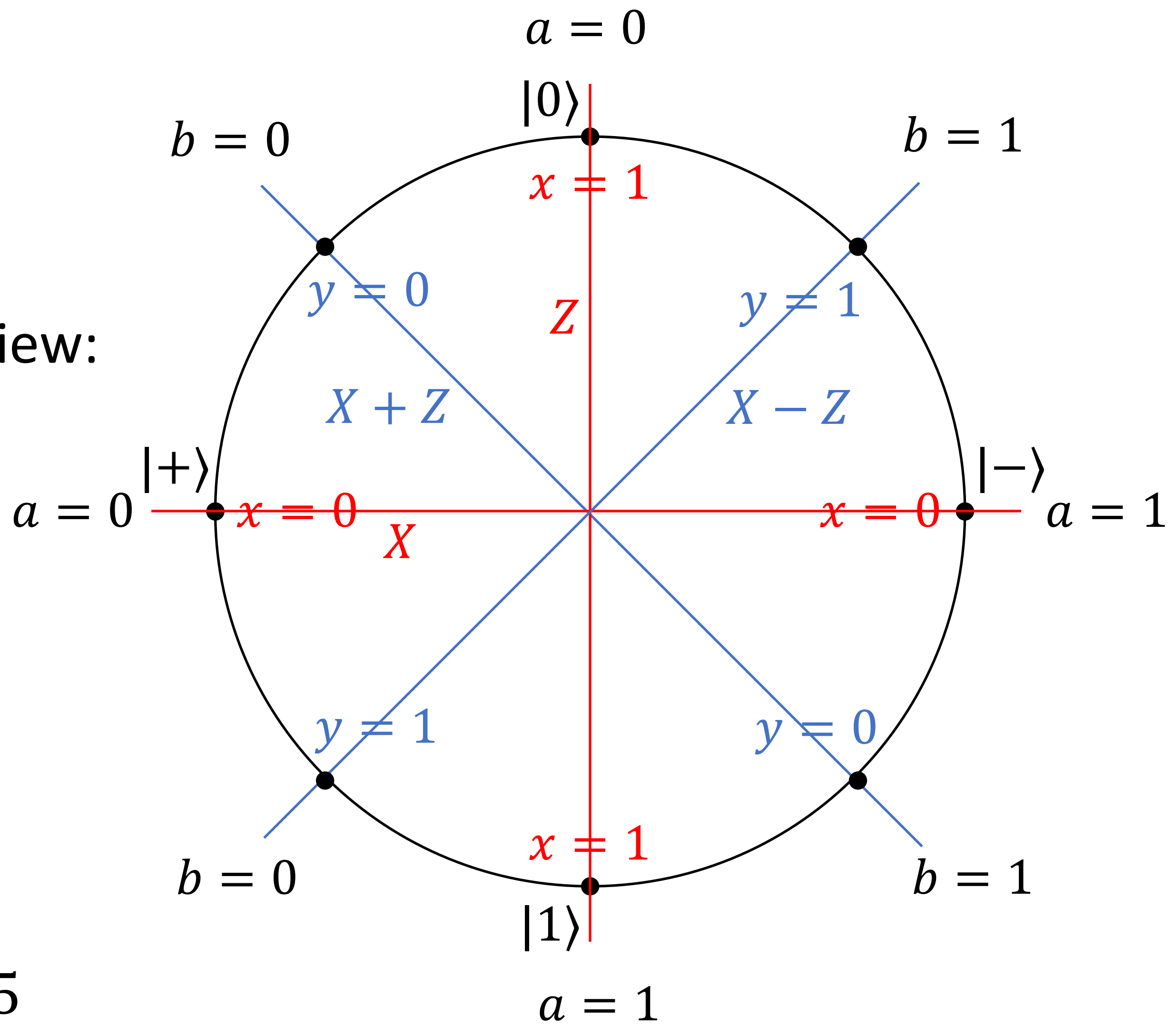
$$|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B = |+\rangle_A|+\rangle_B + |-\rangle_A|-\rangle_B$$



Sample $x, y \leftarrow \{0,1\}$

They win if $a \oplus b = x \wedge y$

Bob's view:



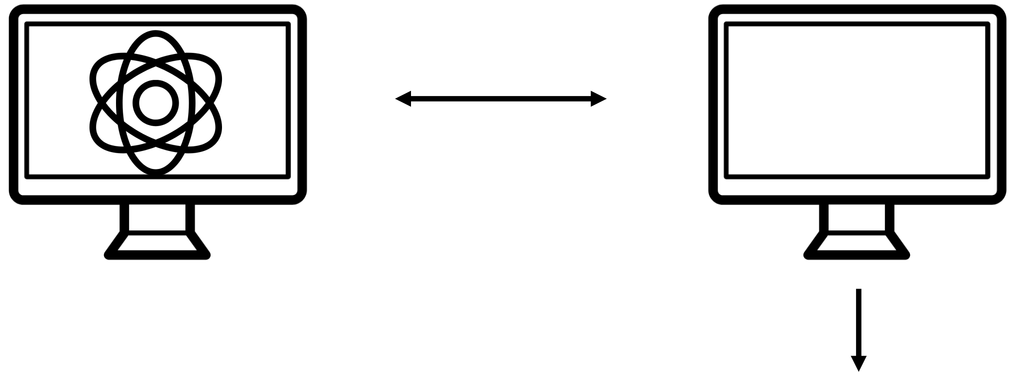
In any case, they win with probability $\approx 0.85 > \omega_{\text{CHSH}}$!

Tsirelson [80]: $\omega^*_{\text{CHSH}} = \cos^2\left(\frac{\pi}{8}\right) \approx 0.85$

From CHSH to proofs of quantumness

- CHSH can be considered a “proof of quantumness” under the assumption that there are two non-communicating provers

- But what about the single prover setting?



Completeness: There is a polynomial-time quantum prover that causes the verifier to accept with probability $v + \epsilon$

Soundness: No polynomial-time classical prover can cause the verifier to accept with probability greater than v

- Shor’s algorithm?

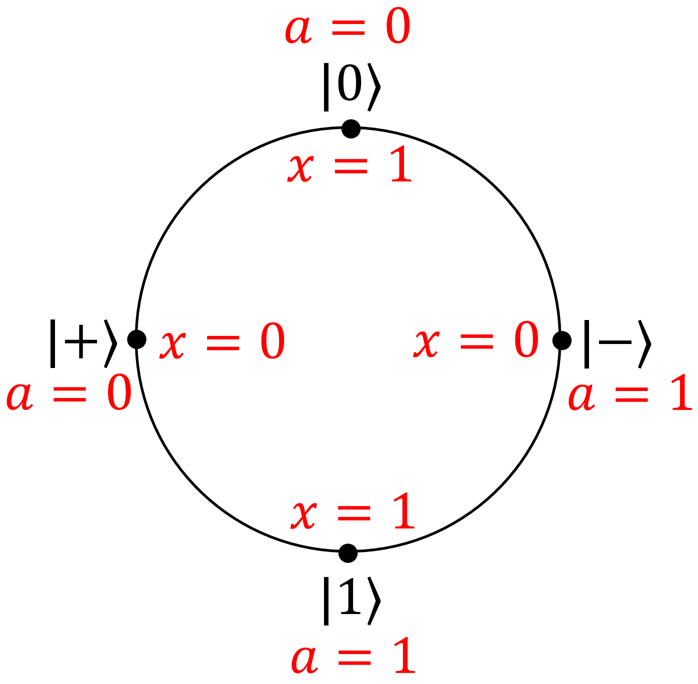
From CHSH to proofs of quantumness

<u>Quantum prover</u>

	$a = 0$	$a = 1$
$x = 0$:	$\vert +\rangle$	$\vert -\rangle$
$x = 1$:	$\vert 0\rangle$	$\vert 1\rangle$

<u>Classical verifier</u>

$x \leftarrow \{0,1\}$



From CHSH to proofs of quantumness

Quantum prover

	$a = 0$	$a = 1$
$x = 0$:	$\|+\rangle$	$\|-\rangle$
$x = 1$:	$\|0\rangle$	$\|1\rangle$

←

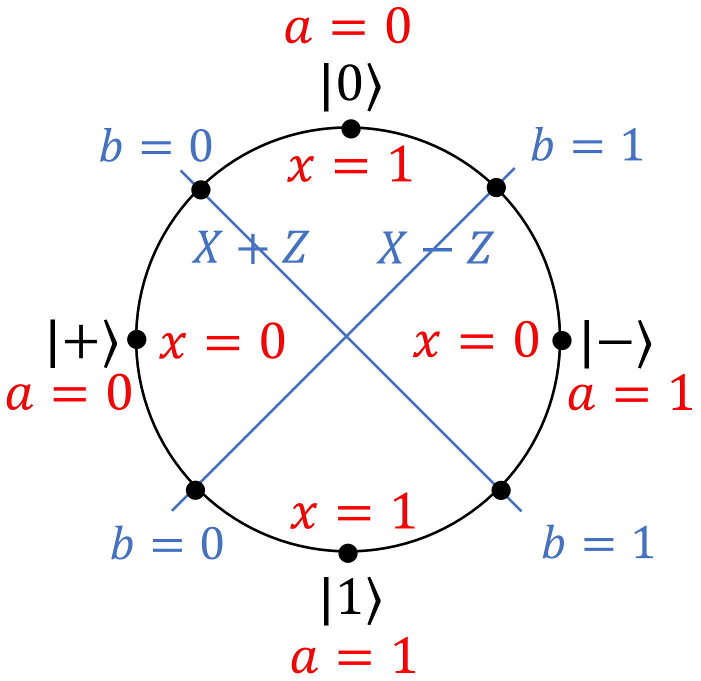
Oblivious BB84 state preparation

Classical verifier

$x \leftarrow \{0,1\}$

← x

a →



$y \leftarrow \{0,1\}$

← y

If $y = 0$, measure $X + Z$
If $y = 1$, measure $X - Z$

b →

Accept if $a \oplus b = x \wedge y$

From CHSH to proofs of quantumness

Quantum prover

	$a = 0$	$a = 1$
$x = 0$:	$\|+\rangle$	$\|-\rangle$
$x = 1$:	$\|0\rangle$	$\|1\rangle$

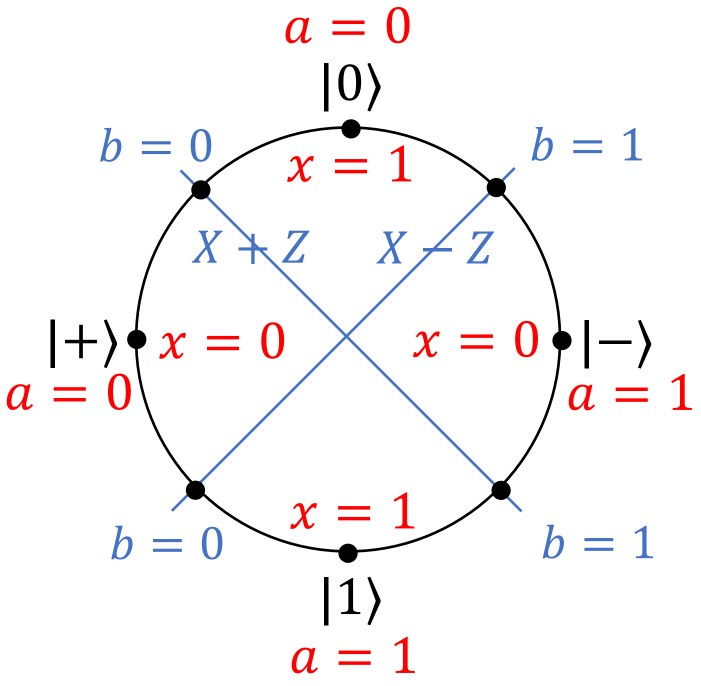
Classical verifier

$x \leftarrow \{0,1\}$

x →

Oblivious BB84 state preparation

← a



Follows from correctness of oblivious BB84 state preparation and the quantum CHSH strategy analysis

If $y = 0$, measure $X + Z$
If $y = 1$, measure $X - Z$

b

Completeness ≈ 0.85

Accept if $a \oplus b = x \wedge y$

From CHSH to proofs of quantumness

Classical prover

	$a = 0$	$a = 1$
$x = 0$:	$\|+\rangle$	$\|-\rangle$
$x = 1$:	$\|0\rangle$	$\|1\rangle$

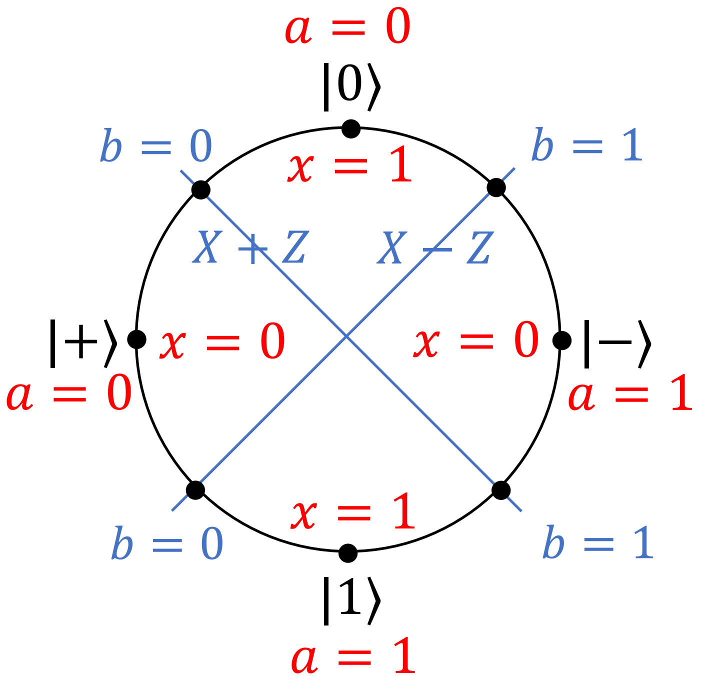
Classical verifier

$x \leftarrow \{0,1\}$

Oblivious BB84 state preparation

x

a



If $y = 0$, measure $X + Z$
If $y = 1$, measure $X - Z$

Follows from correctness of oblivious BB84 state preparation and the quantum CHSH strategy analysis

Completeness ≈ 0.85

Follows from security of oblivious BB84 state preparation (prover can't guess x) and classical CHSH strategy analysis

Soundness ≈ 0.75

From CHSH to proofs of quantumness

[KCVY21]
[AMMW22]
[ABCC24]

Quantum prover | **Classical verifier**

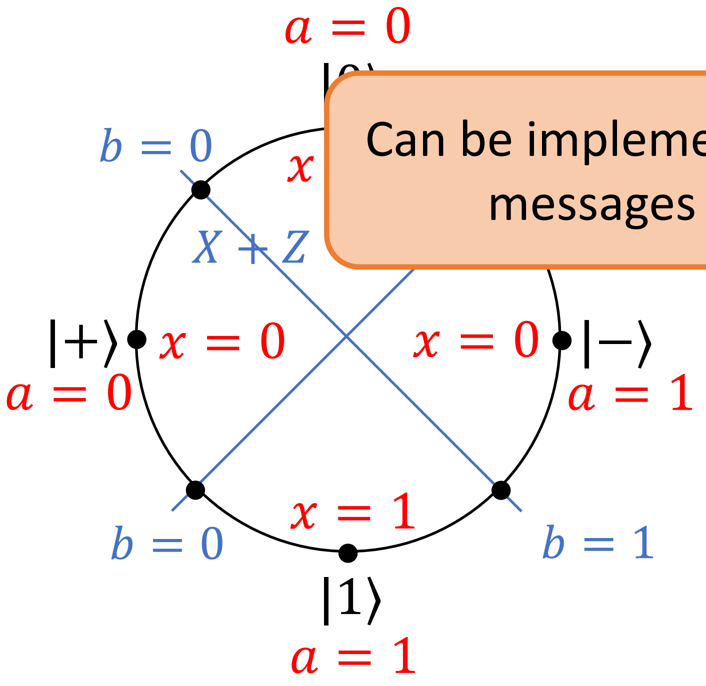
	$a = 0$	$a = 1$
$x = 0$:	$\|+\rangle$	$\|-\rangle$
$x = 1$:	$\|0\rangle$	$\|1\rangle$

Oblivious BB84 state preparation

x ← verifier: $x \leftarrow \{0,1\}$

a → verifier

Can be implemented in two classical messages using any dTCF



y ← verifier: $y \leftarrow \{0,1\}$

If $y = 0$, measure $X + Z$
If $y = 1$, measure $X - Z$

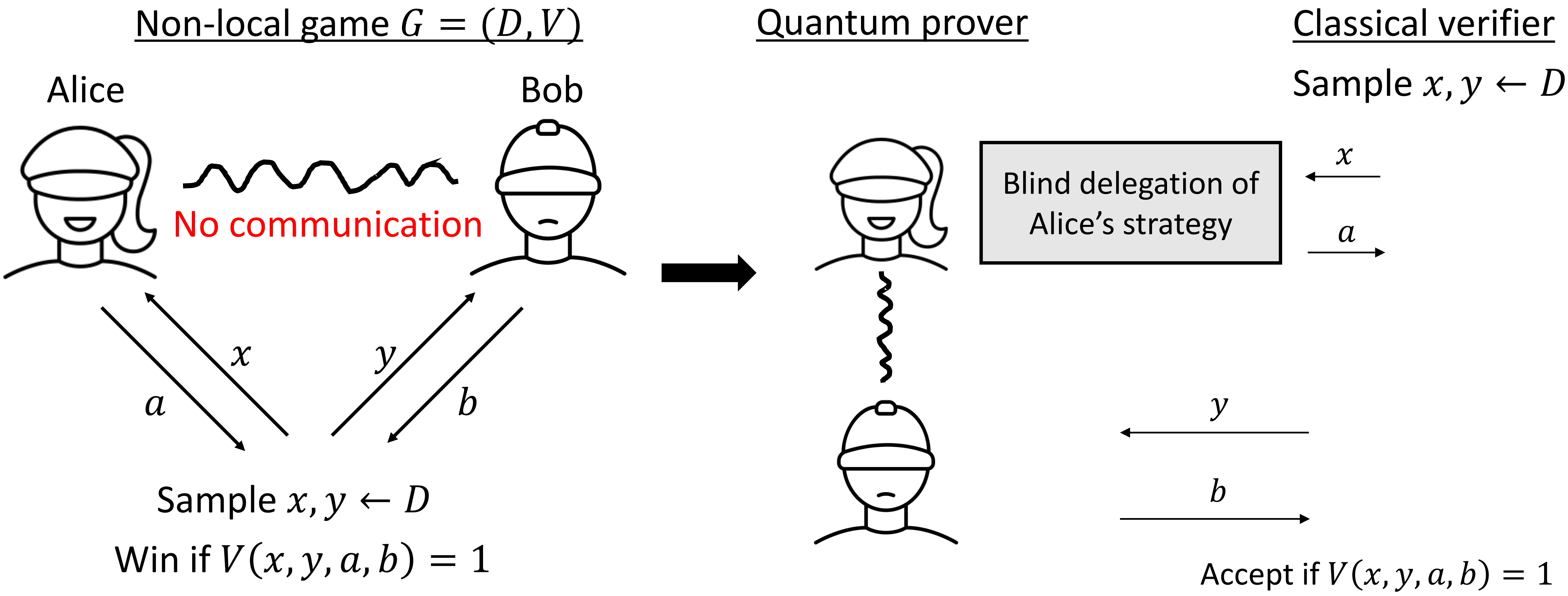
b → verifier

Completeness ≈ 0.85

Soundness ≈ 0.75

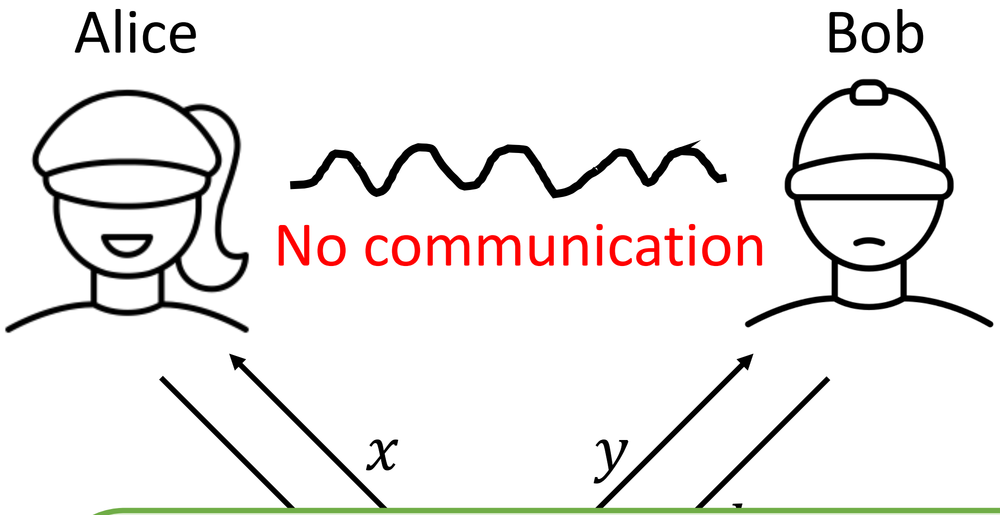
Accept if $a \oplus b = x \wedge y$

Generalization: The KLVY Compiler



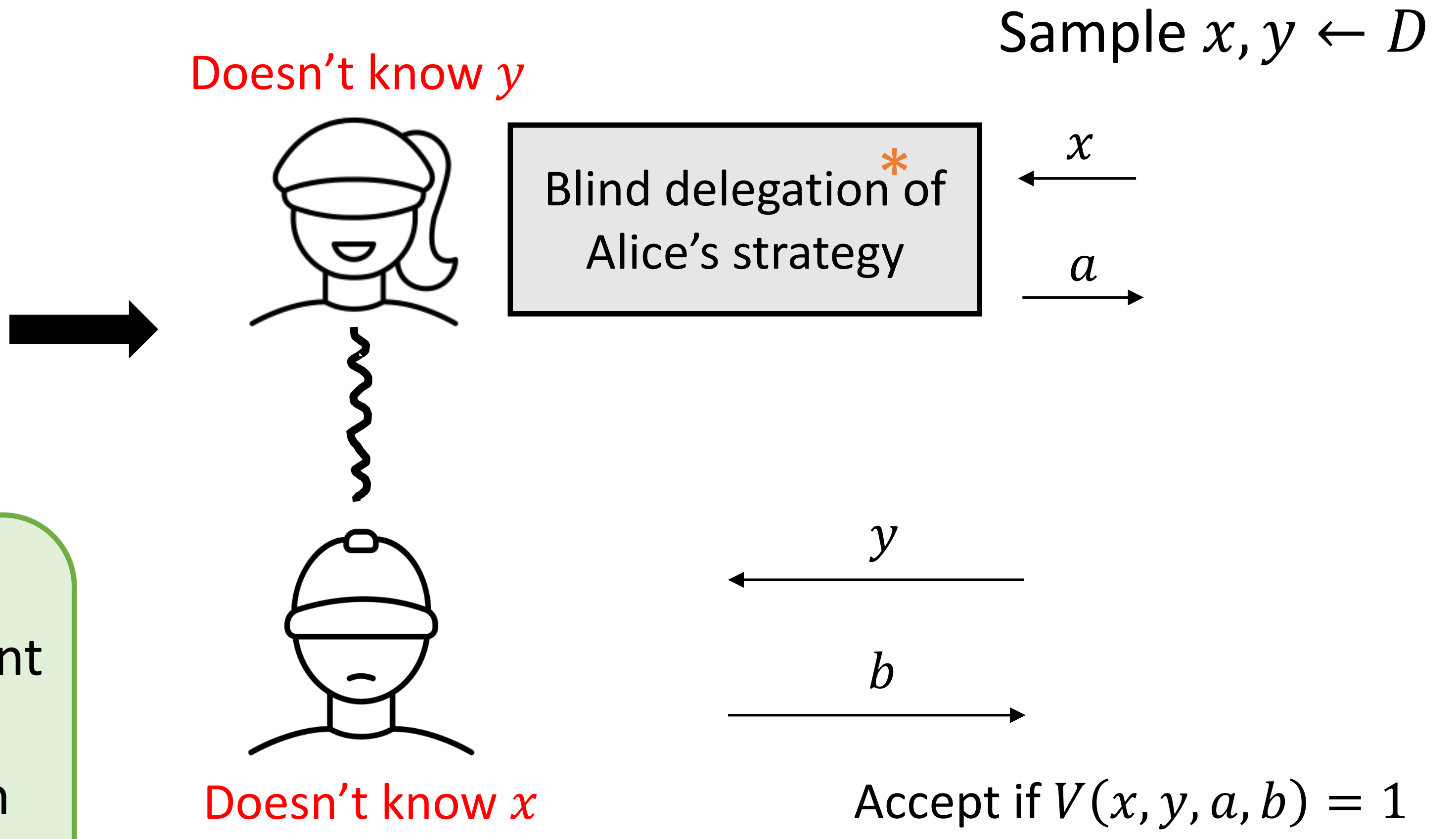
Generalization: The KLVY Compiler

Non-local game $G = (D, V)$



Quantum prover

Classical verifier



[KLVY22] showed:

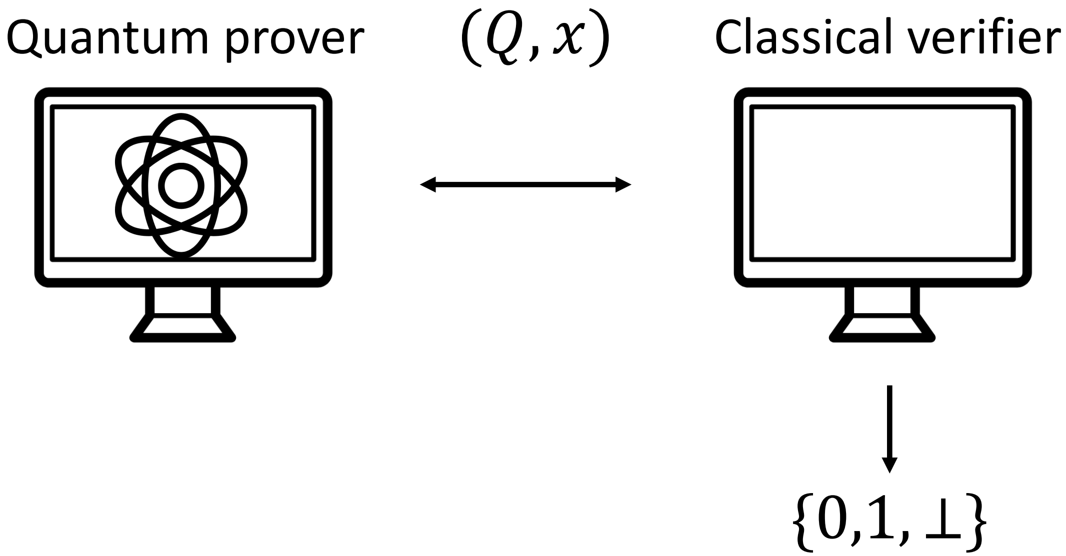
- A QPT quantum prover can implement any QPT two-prover strategy for G
- Any PPT classical prover can win with probability at most $\approx \omega_G$

*[KLVY22] considered only two-message protocols (QFHE)

Verifiable Delegation

- We already had (very simple) proofs of quantumness using the CHSH game, so what was the point of this generalization?

- One reason: can we go beyond proofs of quantumness to classical verification of quantum computation?



Completeness: if prover is honest, output $Q(x)$

Soundness: no QPT malicious prover can force an output of $1 - Q(x)$

Verifiable Delegation

- [RUV13], …, [Gri17], …: Given any BQP computation $Q(x)$, there exists a non-local game G and $\epsilon = 1/\mathrm{poly}$ such that:
  - If $Q(x) = 0$, then $\omega_G^* \geq v + \epsilon$
  - If $Q(x) = 1$, then $\omega_G^* \leq v$

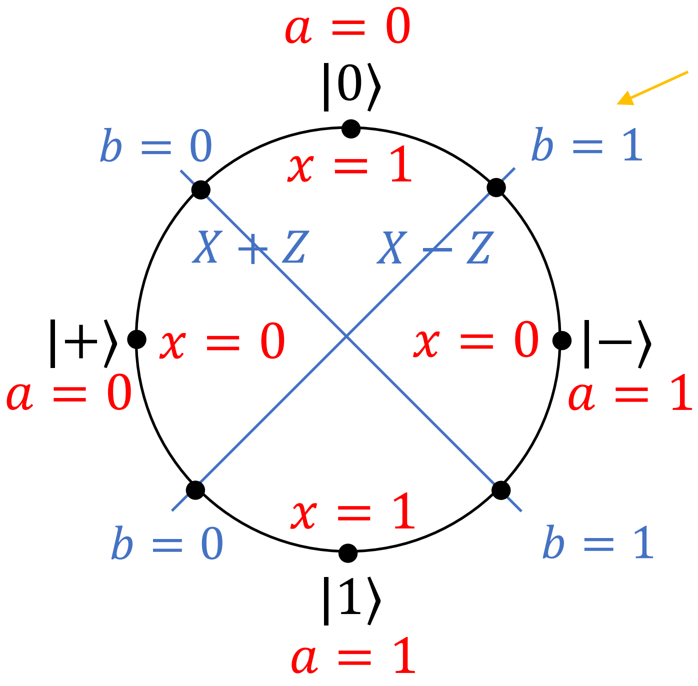
- For proofs of quantumness, we only needed the fact that KLVY preserves the classical value ω_G of the game, since we only care about soundness against classical provers

- For verifiable delegation, we need soundness against quantum provers, and thus have to think about whether the KLVY compiler preserves the *quantum* value ω_G^*

Back to the compiled CHSH game

Quantum prover

	$a = 0$	$a = 1$
$x = 0$:	$\|+\rangle$	$\|-\rangle$
$x = 1$:	$\|0\rangle$	$\|1\rangle$



Classical verifier

$x \leftarrow \{0,1\}$

x →

Oblivious BB84 state preparation

a →

"Rigidity": In order to achieve 0.85, the prover's measurements must be at a maximum angle

Verifier can test that the prover is applying (rotated) standard and Hadamard basis measurements

$y \leftarrow \{0,1\}$

y

If $y = 0$, measure $X + Z$
If $y = 1$, measure $X - Z$

b

Can a malicious quantum prover do any better than 0.85?

Accept if $a \oplus b = x \wedge y$

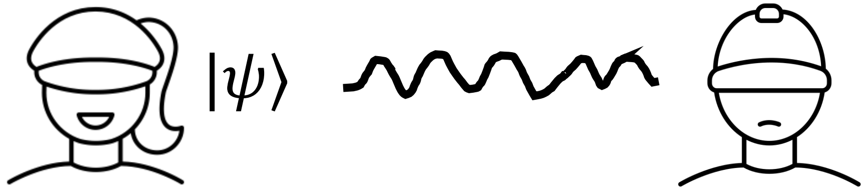
[BGKPV23, NZ23]: No!

Verifiable delegation

How do the [RUV13],[Gri17] non-local games work?

- Ingredient #1: Circuit-to-Hamiltonian
  - $Q, x \rightarrow H_{Q,x} = \sum_i H_i$, where each H_i contains only X or Z terms
  - If $Q(x) = 0$, $\exists |\psi\rangle$ s.t. $\langle\psi|H_{Q,x}|\psi\rangle \geq v + \epsilon$
  - If $Q(x) = 1$, $\forall |\psi\rangle$, $\langle\psi|H_{Q,x}|\psi\rangle \leq v$

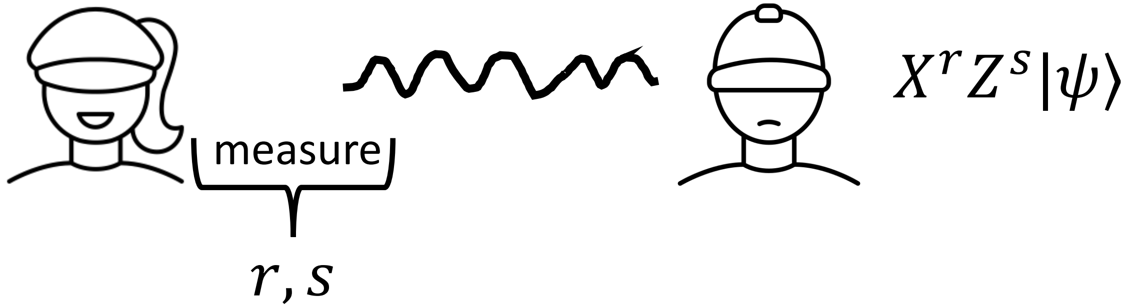
- Ingredient #2: Quantum teleportation



Verifiable delegation

How do the [RUV13],[Gri17] non-local games work?

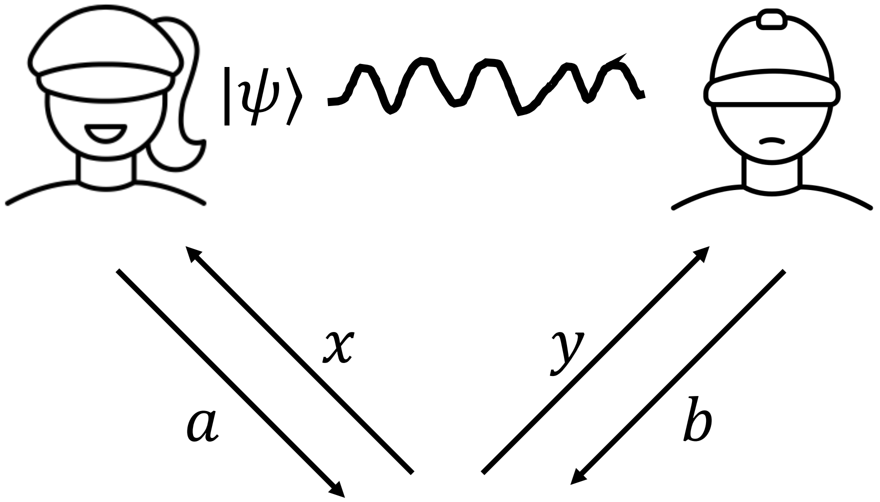
- Ingredient #1: Circuit-to-Hamiltonian
  - $Q, x \rightarrow H_{Q,x} = \sum_i H_i$, where each H_i contains only X or Z terms
  - If $Q(x) = 0$, $\exists|\psi\rangle$ s.t. $\langle\psi|H_{Q,x}|\psi\rangle \geq v + \epsilon$
  - If $Q(x) = 1$, $\forall|\psi\rangle$, $\langle\psi|H_{Q,x}|\psi\rangle \leq v$
- Ingredient #2: Quantum teleportation



- Ingredient #3: Rigidity

Verifiable delegation: Highly simplified

<u>Non-local game for Q, x</u>



Sample $g \leftarrow \{\text{Hamiltonian, CHSH}\}$

Either standard or Hadamard basis measurements

If $g = \text{Ham}$: $x = \text{Tel}$, $a = (r, s)$, $y = H_i$, $b = \langle\psi|X^r Z^s H_i Z^s X^r|\psi\rangle$
accept if average of measurement results $\geq v + \epsilon$

If $g = \text{CHSH}$: play many copies of CHSH game
accept if average win probability ≈ 0.85

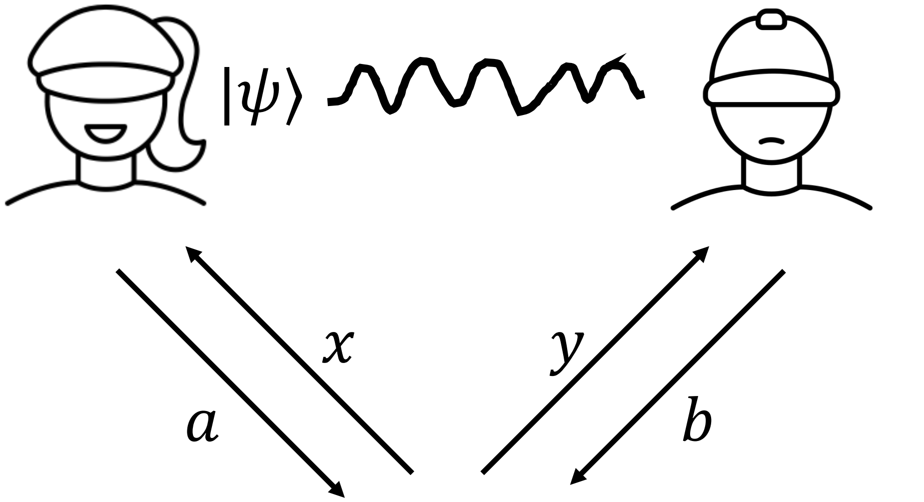
Ensures that Bob is honestly performing the standard and Hadamard basis measurements

Can only win if $|\psi\rangle$ is a valid witness that $Q(x) = 0$

Verifiable delegation: Highly simplified

*[NZ23] considered only two-message protocols (QFHE)

<u>Non-local game for Q, x</u>



Sample $g \leftarrow \{\text{Hamiltonian}, \text{CHSH}\}$

If $g = \text{Ham}$: $x = \text{Tel}$, $a = (r, s)$, $y = H_i$, $b = \langle\psi|X^r Z^s H_i Z^s X^r|\psi\rangle$
accept if average of measurement results $\geq v + \epsilon$

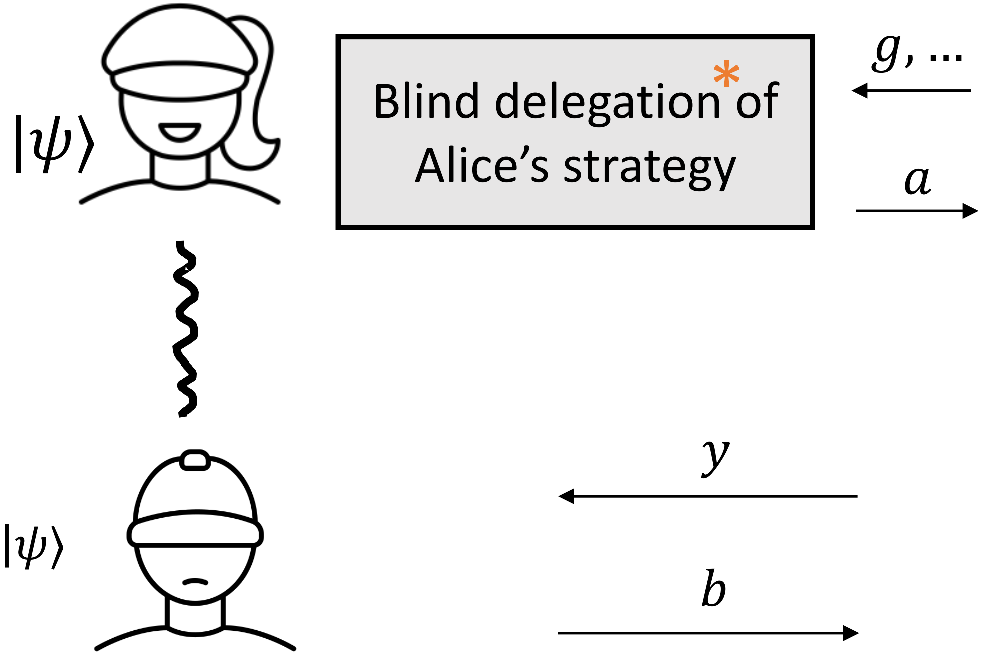
If $g = \text{CHSH}$: play many copies of CHSH game
accept if average win probability ≈ 0.85

Can only win if $|\psi\rangle$ is a valid witness that $Q(x) = 0$

[NZ23] →

<u>Quantum prover</u> Q, x <u>Classical verifier</u>

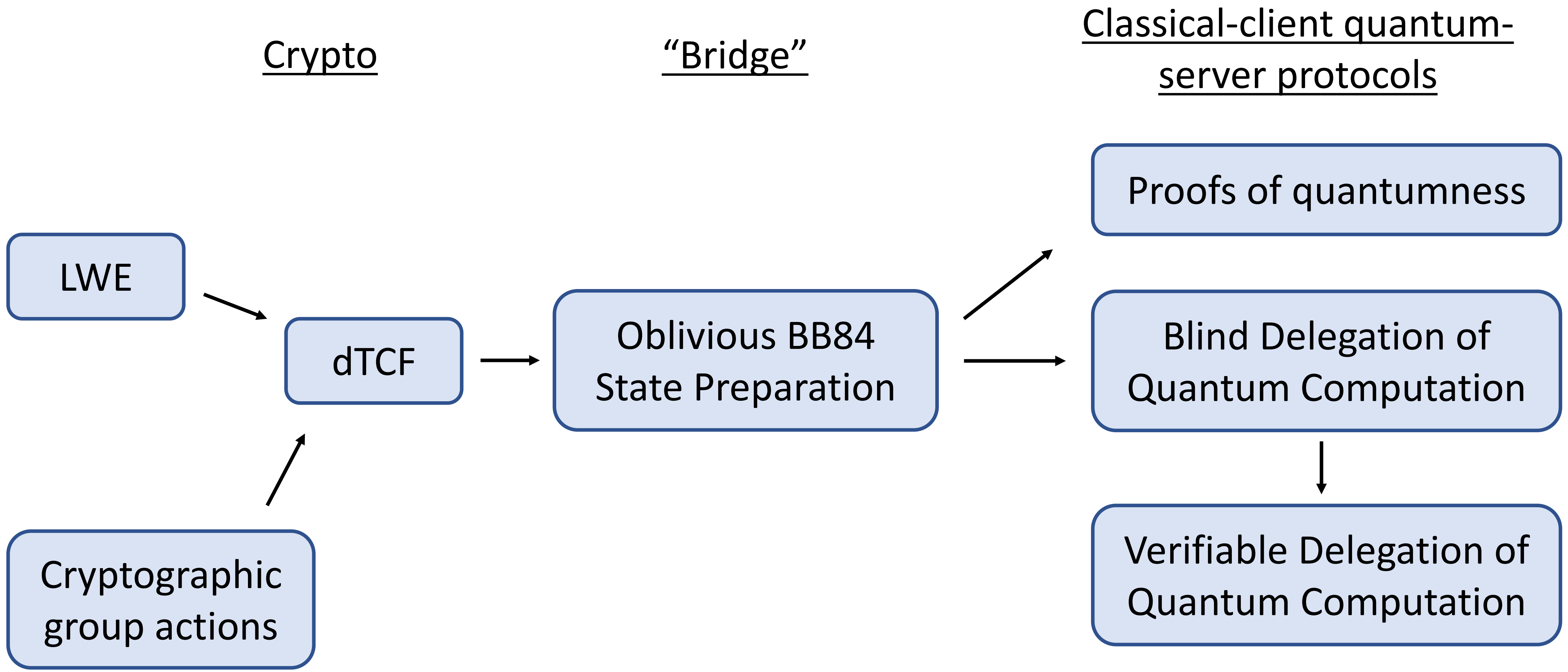
Sample $g \leftarrow \{\text{Ham}, \text{CHSH}\}$



Key: Doesn't know which game is being played

Will only accept if $Q(x) = 0$

Recap


Crypto
"Bridge"
Classical-client quantum-server protocols
LWE
dTCF
Cryptographic group actions
Oblivious BB84 State Preparation
Proofs of quantumness
Blind Delegation of Quantum Computation
Verifiable Delegation of Quantum Computation

Key References

- Blind quantum computation with a weak quantum client
  - Andrew Childs. *Secure Assisted Quantum Computation.* 2001. https://arxiv.org/abs/quant-ph/0111046
  - Anne Broadbent. *Delegating Private Quantum Computations.* 2015. https://arxiv.org/abs/1506.01328

- Introducing trapdoor claw-free functions
  - Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, Thomas Vidick. *A Cryptographic Test of Quantumness and Certifiable Randomness from a Single Quantum Device.* 2018. https://arxiv.org/abs/1804.00640
  - Urmila Mahadev. *Classical Homomorphic Encryption for Quantum Circuits.* 2017. https://arxiv.org/abs/1708.02130\
  - Urmila Mahadev. *Classical Verification of Quantum Computations.* 2018. https://arxiv.org/abs/1804.01082
  - Alexandru Cojocaru, Léo Colisson, Elham Kashefi, Petros Wallden. *QFactory: Classically-Instructed Remote Secret Qubits Preparation.* 2019. https://arxiv.org/pdf/1904.06303

- The non-local game approach
  - Gregory Kahanamoku-Meyer, Soonwon Choi, Umesh Vazirani, Norman Yao. *Classically-Verifiable Quantum Advantage from a Computational Bell Test.* 2021. https://arxiv.org/abs/2104.00687
  - Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, Lisa Yang. *Quantum Advantage from Any Non-Local Game.* 2022. https://arxiv.org/abs/2203.15877
  - Zvika Brakerski, Alexandru Georghiu, Gregory Kahanamoku-Meyer, Eitan Porat, Thomas Vidick. *Simple Tests of Quantumness also Certify Qubits.* 2023. https://arxiv.org/abs/2303.01293
  - Anand Natarajan, Tina Zhang. *Bounding the quantum value of compiled nonlocal games: from CHSH to BQP verification.* 2023. https://arxiv.org/abs/2303.01545