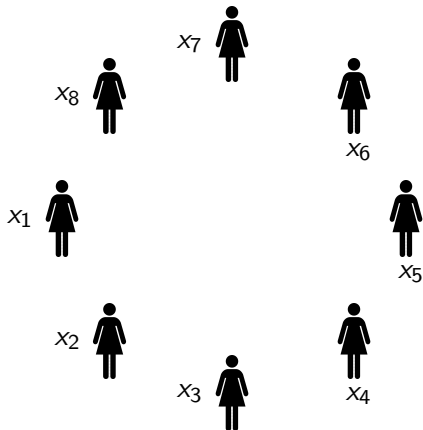


Multi-party computation in the quantum world

Alex Bredariol Grilo

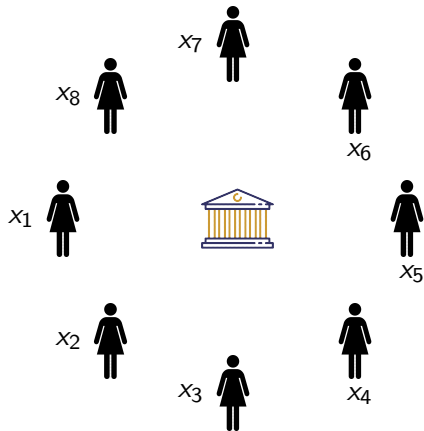


Multi-party computation



Goal: Compute $f(x_1, \dots, x_8)$ without revealing their input

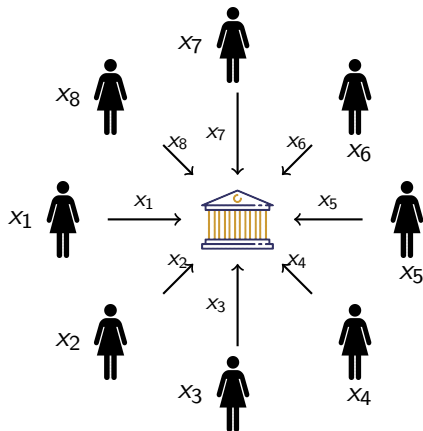
Multi-party computation



Goal: Compute $f(x_1, \dots, x_8)$ without revealing their input

Ideal world

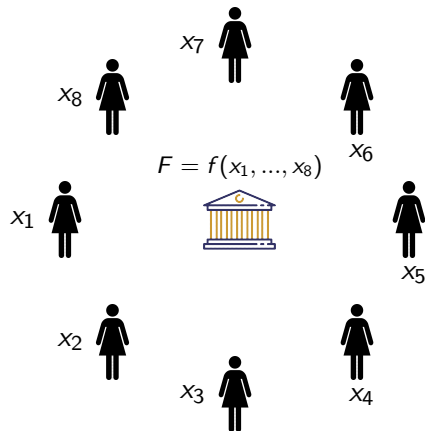
Multi-party computation



Goal: Compute $f(x_1, \dots, x_8)$ without revealing their input

Ideal world

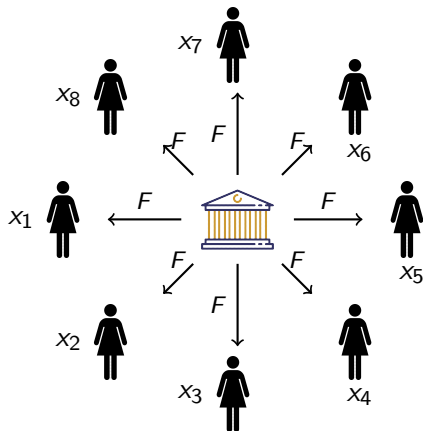
Multi-party computation



Goal: Compute $f(x_1, \dots, x_8)$ without revealing their input

Ideal world

Multi-party computation

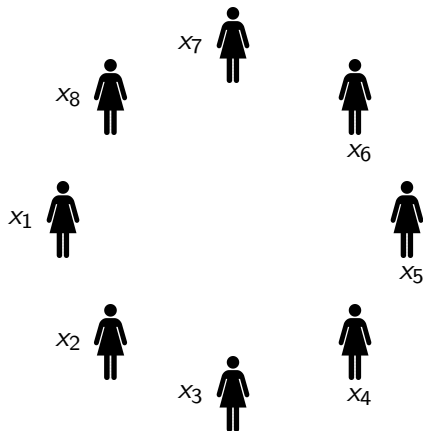


Goal: Compute $f(x_1, \dots, x_8)$ without revealing their input

Ideal world

- Each party learns $F = f(x_1, \dots, x_8)$ and nothing else

Multi-party computation



Goal: Compute $f(x_1, \dots, x_8)$ without revealing their input

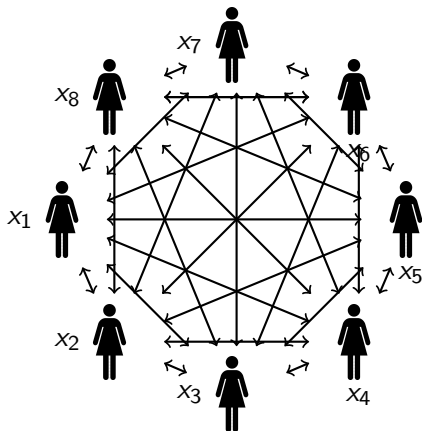
Ideal world

- Each party learns $F = f(x_1, \dots, x_8)$ and nothing else

Real world

- Goal: implement the ideal functionality

Multi-party computation



Goal: Compute $f(x_1, \dots, x_8)$ without revealing their input

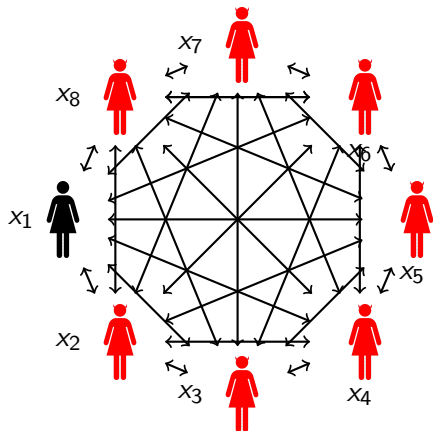
Ideal world

- Each party learns $F = f(x_1, \dots, x_8)$ and nothing else

Real world

- Goal: implement the ideal functionality
- Protocols where parties interact, but still they only learn F

Multi-party computation



Goal: Compute $f(x_1, \dots, x_8)$ without revealing their input

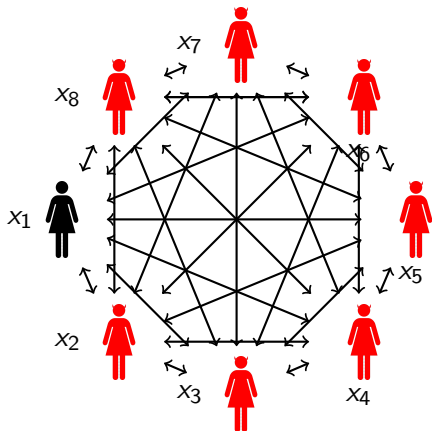
Ideal world

- Each party learns $F = f(x_1, \dots, x_8)$ and nothing else

Real world

- Goal: implement the ideal functionality
- Protocols where parties interact, but still they only learn F
- Even if they behave dishonestly

Multi-party computation



Goal: Compute $f(x_1, \dots, x_8)$ without revealing their input

Ideal world

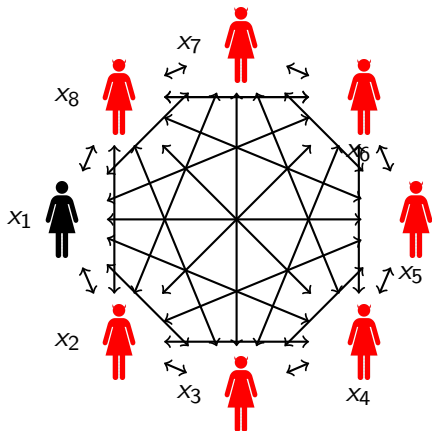
- Each party learns $F = f(x_1, \dots, x_8)$ and nothing else

Real world

- Goal: implement the ideal functionality
- Protocols where parties interact, but still they only learn F
- Even if they behave dishonestly

Are classical protocols
secure against quantum
adversaries?

Multi-party computation



Goal: Compute $f(x_1, \dots, x_8)$ without revealing their input

Ideal world

- Each party learns $F = f(x_1, \dots, x_8)$ and nothing else

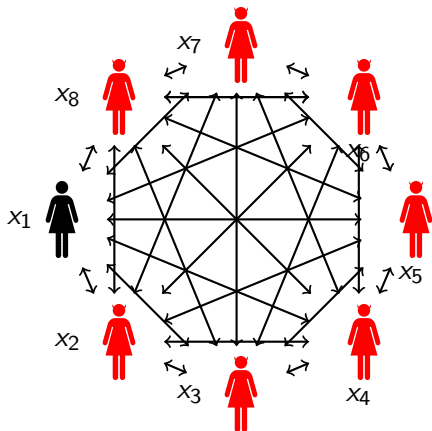
Real world

- Goal: implement the ideal functionality
- Protocols where parties interact, but still they only learn F
- Even if they behave dishonestly

Are classical protocols
secure against quantum
adversaries?

Are there *better* quantum
protocols for MPC?

Multi-party computation



Goal: Compute $f(x_1, \dots, x_8)$ without revealing their input

Ideal world

- Each party learns $F = f(x_1, \dots, x_8)$ and nothing else

Real world

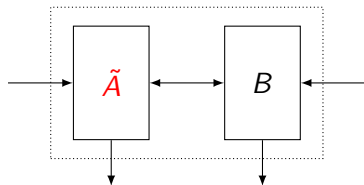
- Goal: implement the ideal functionality
- Protocols where parties interact, but still they only learn F
- Even if they behave dishonestly

Are classical protocols
secure against quantum
adversaries?

Are there *better* quantum
protocols for MPC?

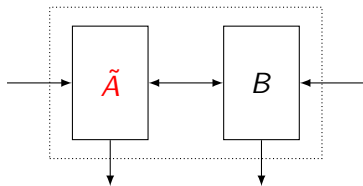
Are there protocols for
MPQC?

Security definition (two-party case)

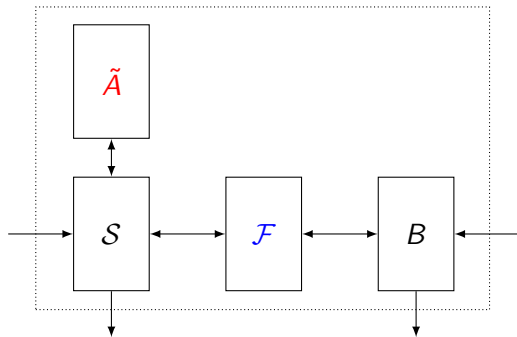


Real world

Security definition (two-party case)

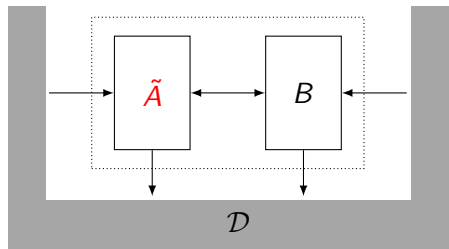


Real world

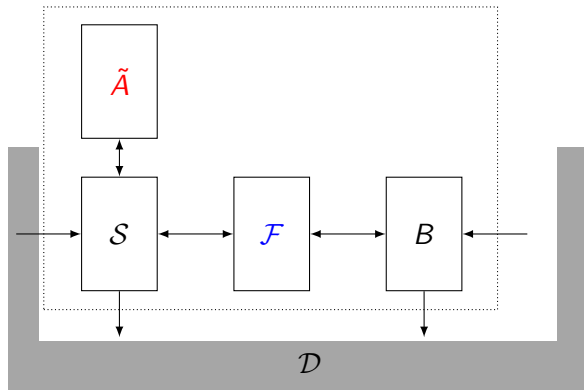


Ideal world

Security definition (two-party case)

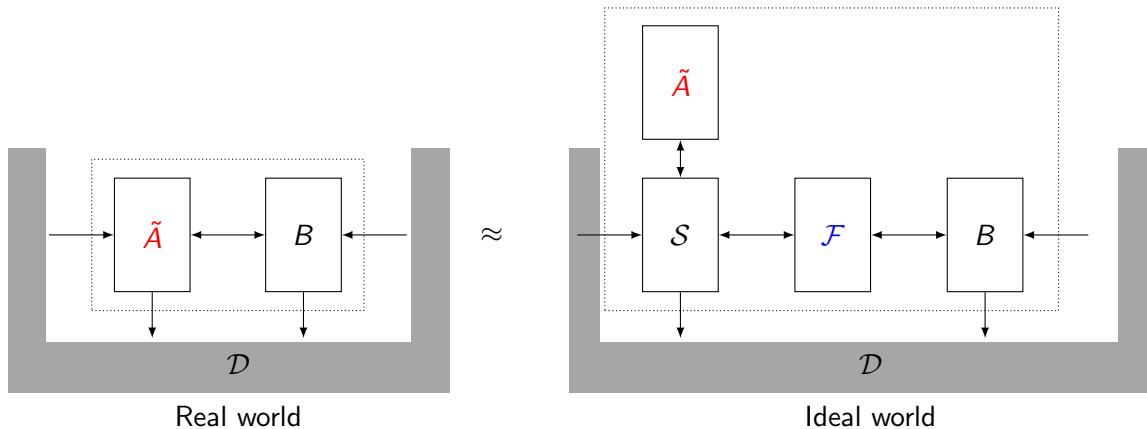


Real world



Ideal world

Security definition (two-party case)



For every polynomial-time $\mathcal{D}$, $|Pr[\mathcal{D}(\text{real})] - Pr[\mathcal{D}(\text{ideal})]| \leq \text{negl}(\lambda)$

Classical MPC protocols

Classical MPC protocols

GMW family

Honest MPC + $\mathcal{F}_{ZK}$

Classical MPC protocols

GMW family

Honest MPC + $\mathcal{F}_{ZK}$

IPS family

Honest MPC + $\mathcal{F}_{OT}$

Classical MPC protocols

GMW family

Honest MPC + $\mathcal{F}_{ZK}$

IPS family

Honest MPC + $\mathcal{F}_{OT}$

- Unruh'10: protocols are secure against quantum adversaries in the ideal world

Classical MPC protocols

GMW family

Honest MPC + $\mathcal{F}_{ZK}$

IPS family

Honest MPC + $\mathcal{F}_{OT}$

- Unruh'10: protocols are secure against quantum adversaries in the ideal world
- Implementation of ideal functionalities
 - ▶ Classically: from PKE assumptions
 - ▶ Quantumly: extraction without disturbing internal state of adversaries is cumbersome

Classical MPC protocols

GMW family

Honest MPC + $\mathcal{F}_{ZK}$

IPS family

Honest MPC + $\mathcal{F}_{OT}$

- Unruh'10: protocols are secure against quantum adversaries in the ideal world
- Implementation of ideal functionalities
 - ▶ Classically: from PKE assumptions
 - ▶ Quantumly: extraction without disturbing internal state of adversaries is cumbersome
- Solutions

Classical MPC protocols

GMW family

Honest MPC + $\mathcal{F}_{ZK}$

IPS family

Honest MPC + $\mathcal{F}_{OT}$

- Unruh'10: protocols are secure against quantum adversaries in the ideal world
- Implementation of ideal functionalities
 - ▶ Classically: from PKE assumptions
 - ▶ Quantumly: extraction without disturbing internal state of adversaries is cumbersome
- Solutions
 - ① Implementations from trusted setup (e.g. Garg-S'18)

Classical MPC protocols

GMW family

Honest MPC + $\mathcal{F}_{ZK}$

IPS family

Honest MPC + $\mathcal{F}_{OT}$

- Unruh'10: protocols are secure against quantum adversaries in the ideal world
- Implementation of ideal functionalities
 - ▶ Classically: from PKE assumptions
 - ▶ Quantumly: extraction without disturbing internal state of adversaries is cumbersome
- Solutions
 - 1 Implementations from trusted setup (e.g. Garg-S'18)
 - 2 Implementations from stronger functionalities/assumptions (Bitansky-S'20, Agarwal-BGKM'20)

Classical MPC protocols

GMW family

Honest MPC + $\mathcal{F}_{ZK}$

IPS family

Honest MPC + $\mathcal{F}_{OT}$

- Unruh'10: protocols are secure against quantum adversaries in the ideal world
- Implementation of ideal functionalities
 - ▶ Classically: from PKE assumptions
 - ▶ Quantumly: extraction without disturbing internal state of adversaries is cumbersome
- Solutions
 - 1 Implementations from trusted setup (e.g. Garg-S'18)
 - 2 Implementations from stronger functionalities/assumptions (Bitansky-S'20, Agarwal-BGKM'20)
 - 3 Implementations with quantum protocols

Classical MPC protocols

GMW family

Honest MPC + $\mathcal{F}_{ZK}$

IPS family

Honest MPC + $\mathcal{F}_{OT}$

- Unruh'10: protocols are secure against quantum adversaries in the ideal world
- Implementation of ideal functionalities
 - ▶ Classically: from PKE assumptions
 - ▶ Quantumly: extraction without disturbing internal state of adversaries is cumbersome
- Solutions
 - 1 Implementations from trusted setup (e.g. Garg-S'18)
 - 2 Implementations from stronger functionalities/assumptions (Bitansky-S'20, Agarwal-BGKM'20)
 - 3 Implementations with quantum protocols (from weaker assumptions!)

Classical MPC protocols

GMW family

Honest MPC + $\mathcal{F}_{ZK}$

IPS family

Honest MPC + $\mathcal{F}_{OT}$

- Unruh'10: protocols are secure against quantum adversaries in the ideal world
- Implementation of ideal functionalities
 - ▶ Classically: from PKE assumptions
 - ▶ Quantumly: extraction without disturbing internal state of adversaries is cumbersome
- Solutions
 - 1 Implementations from trusted setup (e.g. Garg-S'18)
 - 2 Implementations from stronger functionalities/assumptions (Bitansky-S'20, Agarwal-BGKM'20)
 - 3 **Implementations with quantum protocols (from weaker assumptions!)**

Oblivious transfer

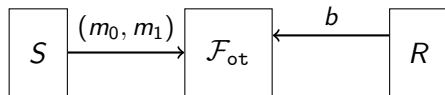
Oblivious transfer

Ideal functionality



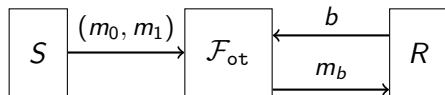
Oblivious transfer

Ideal functionality



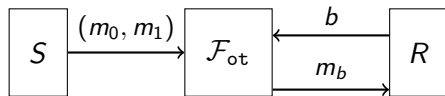
Oblivious transfer

Ideal functionality

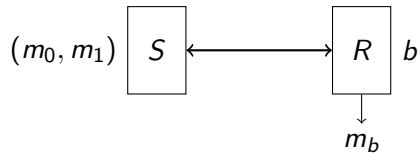


Oblivious transfer

Ideal functionality



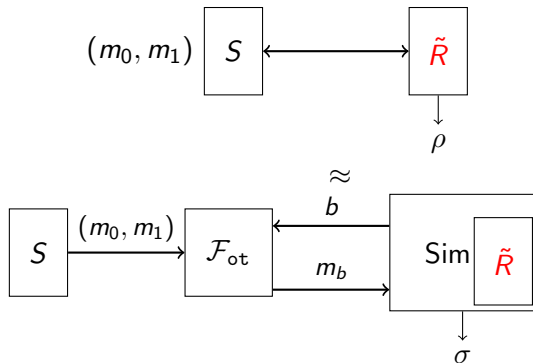
Real world



Oblivious transfer - security definitions

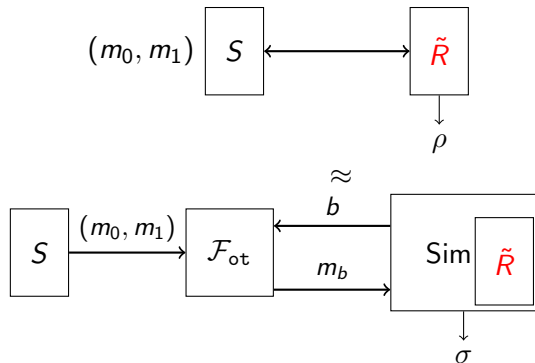
Oblivious transfer - security definitions

Security against malicious receiver

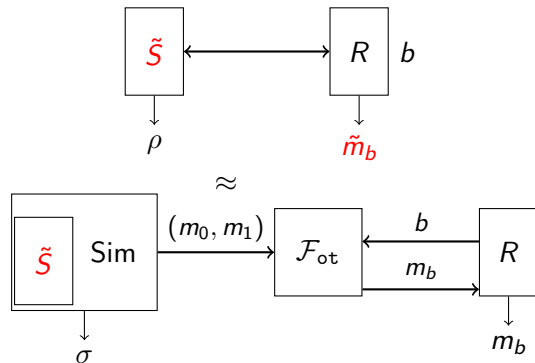


Oblivious transfer - security definitions

Security against malicious receiver



Security against malicious sender



MPC from Quantum+OWF

- IPS'08: MPC protocols from $\mathcal{F}_{ot}$

MPC from Quantum+OWF

- IPS'08: MPC protocols from $\mathcal{F}_{ot}$
- Unruh'10: Classical reduction from $\mathcal{F}_{ot}$ to MPC holds in the quantum world

MPC from Quantum+OWF

- IPS'08: MPC protocols from $\mathcal{F}_{ot}$
- Unruh'10: Classical reduction from $\mathcal{F}_{ot}$ to MPC holds in the quantum world
- CK'88,BBCS'92: Quantum protocol for OT based on commitment schemes

MPC from Quantum+OWF

- IPS'08: MPC protocols from $\mathcal{F}_{ot}$
- Unruh'10: Classical reduction from $\mathcal{F}_{ot}$ to MPC holds in the quantum world
- CK'88,BBCS'92: Quantum protocol for OT based on commitment schemes
- DFLSS'09 BF'10: Security proof of CK/BBCS protocol based on strong classical commitment schemes (likely to be stronger than OWF)

MPC from Quantum+OWF

- IPS'08: MPC protocols from $\mathcal{F}_{ot}$
- Unruh'10: Classical reduction from $\mathcal{F}_{ot}$ to MPC holds in the quantum world
- CK'88,BBCS'92: Quantum protocol for OT based on commitment schemes
- DFLSS'09 BF'10: Security proof of CK/BBCS protocol based on strong classical commitment schemes (likely to be stronger than OWF)
- BCKM'21 and GLSV'21: Quantum protocol for strong commitment from OWF

MPC from Quantum+OWF

- IPS'08: MPC protocols from $\mathcal{F}_{ot}$
- Unruh'10: Classical reduction from $\mathcal{F}_{ot}$ to MPC holds in the quantum world
- CK'88,BBCS'92: Quantum protocol for OT based on commitment schemes
- DFLSS'09 BF'10: Security proof of CK/BBCS protocol based on strong classical commitment schemes (likely to be stronger than OWF)
- BCKM'21 and GLSV'21: Quantum protocol for strong commitment from OWF

Corollary

Quantum protocol for MPC from OWF

MPC from Quantum+OWF

- IPS'08: MPC protocols from $\mathcal{F}_{ot}$
- Unruh'10: Classical reduction from $\mathcal{F}_{ot}$ to MPC holds in the quantum world
- CK'88,BBCS'92: Quantum protocol for OT based on commitment schemes
- DFLSS'09 BF'10: Security proof of CK/BBCS protocol based on strong classical commitment schemes (likely to be stronger than OWF)
- BCKM'21 and GLSV'21: Quantum protocol for strong commitment from OWF

Corollary

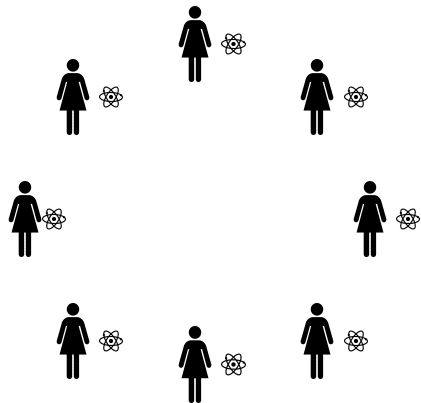
Quantum protocol for MPC from OWF

vs.

IR'91: We don't expect classical MPC from OWF!

Multi-party quantum computation

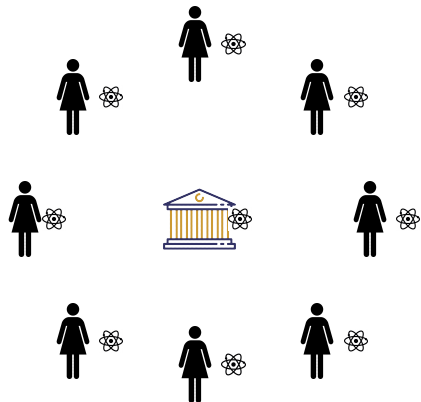
Parties share some input state $\rho_{A_1 \dots A_8}$



Goal: Compute U on joint share state ρ without revealing their share

Multi-party quantum computation

Parties share some input state $\rho_{A_1 \dots A_8}$

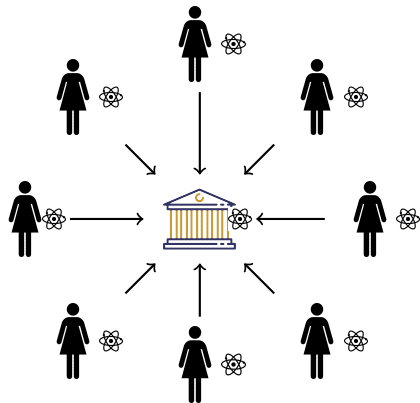


Goal: Compute U on joint share state ρ without revealing their share

Ideal world

Multi-party quantum computation

Parties share some input state $\rho_{A_1 \dots A_8}$

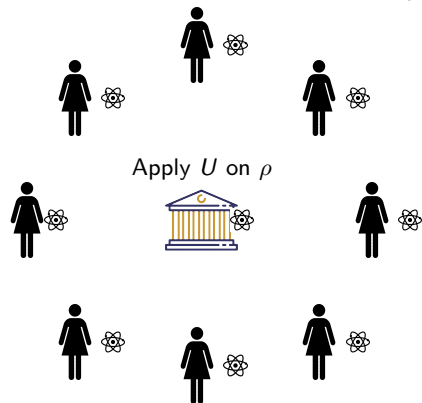


Goal: Compute U on joint share state ρ without revealing their share

Ideal world

Multi-party quantum computation

Parties share some input state $\rho_{A_1 \dots A_8}$

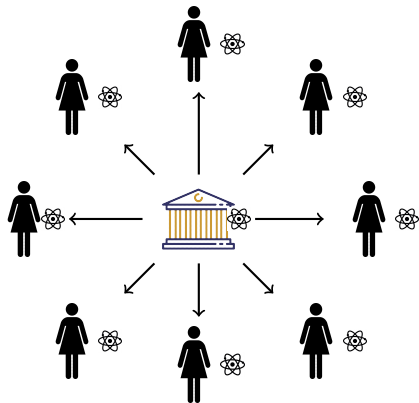


Goal: Compute U on joint share state ρ without revealing their share

Ideal world

Multi-party quantum computation

Parties share some input state $\rho_{A_1 \dots A_8}$



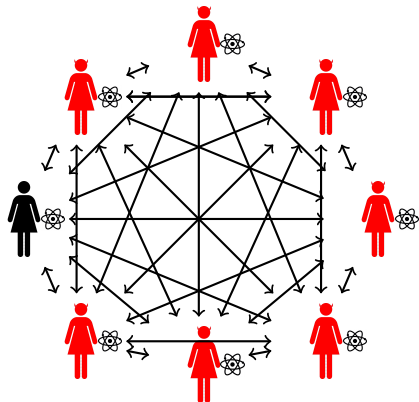
Goal: Compute U on joint share state ρ without revealing their share

Ideal world

- Each party gets their share of the output $U\rho U$

Multi-party quantum computation

Parties share some input state $\rho_{A_1 \dots A_8}$



Goal: Compute U on joint share state ρ without revealing their share

Ideal world

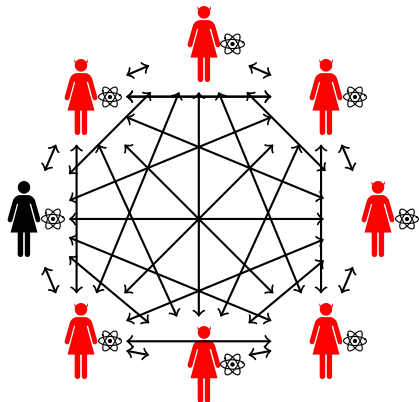
- Each party gets their share of the output $U\rho U$

Real world

- Goal: implement the ideal functionality
- Protocols where parties interact, but still they only learn their share even if they behave dishonestly

Multi-party quantum computation

Parties share some input state $\rho_{A_1 \dots A_8}$



Goal: Compute U on joint share state ρ without revealing their share

Ideal world

- Each party gets their share of the output $U\rho U$

Real world

- Goal: implement the ideal functionality
- Protocols where parties interact, but still they only learn their share even if they behave dishonestly
- Security definition similar to the classical setting

MPQC

- Statistically secure MPQC with honest majority [CGS'02,BCGHS'06]
- Computationally secure 2PQC [DNS'10, DNS'12, KMW'17]
- MPQC with allowed dishonest subsets [KP'17]
- Computationally secure MPQC with arbitrary dishonest majority [DGJMS'20]

MPQC

- Statistically secure MPQC with honest majority [CGS'02,BCGHS'06]
- Computationally secure 2PQC [DNS'10, DNS'12, KMW'17]
- MPQC with allowed dishonest subsets [KP'17]
- **Computationally secure MPQC with arbitrary dishonest majority** [D**G**JMS'20]

MPQC

- Statistically secure MPQC with honest majority [CGS'02,BCGHS'06]
- Computationally secure 2PQC [DNS'10, DNS'12, KMW'17]
- MPQC with allowed dishonest subsets [KP'17]
- **Computationally secure MPQC with arbitrary dishonest majority** [DGJMS'20]
 - ▶ Extends DNS'12 to the multi-party setting
 - ▶ Assumes ideal MPC functionality ($\mathcal{F}_{MPC}$)

Clifford encoding

Clifford encoding

Clifford operations:

Unitaries generated by $\{H, P, CNOT\}$

$\mathcal{C}_m = \{\text{Clifford circuits on } m \text{ qubits}\}$

Clifford encoding

Clifford operations:

Unitaries generated by $\{H, P, CNOT\}$

$\mathcal{C}_m = \{\text{Clifford circuits on } m \text{ qubits}\}$

Clifford encoding for n -qubit state $|\psi\rangle$ and security parameter λ :

- 1 Pick a random $(\lambda + n)$ -qubit Clifford C
- 2 $C(|\psi\rangle \otimes |0^\lambda\rangle)$

Clifford encoding

Clifford operations:

Unitaries generated by $\{H, P, CNOT\}$

$\mathcal{C}_m = \{\text{Clifford circuits on } m \text{ qubits}\}$

Clifford encoding for n -qubit state $|\psi\rangle$ and security parameter λ :

- 1 Pick a random $(\lambda + n)$ -qubit Clifford C
- 2 $C(|\psi\rangle \otimes |0^\lambda\rangle)$

Privacy:

$C|\psi\rangle$ is one-time padded

Clifford encoding

Clifford operations:

Unitaries generated by $\{H, P, CNOT\}$

$\mathcal{C}_m = \{\text{Clifford circuits on } m \text{ qubits}\}$

Clifford encoding for n -qubit state $|\psi\rangle$ and security parameter λ :

- 1 Pick a random $(\lambda + n)$ -qubit Clifford C
- 2 $C(|\psi\rangle \otimes |0^\lambda\rangle)$

Privacy:

$C|\psi\rangle$ is one-time padded

Authentication:

For any non-trivial $\mathcal{A}$, trap qubits of $C^\dagger \mathcal{A}(C|\psi\rangle|0^n\rangle)$ will be non-zero w.p. $1 - \text{negl}(\lambda)$

MPQC protocol - General idea

Focus on a single (pure) qubit

MPQC protocol - General idea

Focus on a single (pure) qubit

- P_{i^*} holds $C(|\psi\rangle|0^{2\lambda}\rangle)$
- All players (secret) share C
 - ▶ Players share random C_i 's s.t. $C_k \dots C_1 = C$

MPQC protocol - General idea

Focus on a single (pure) qubit

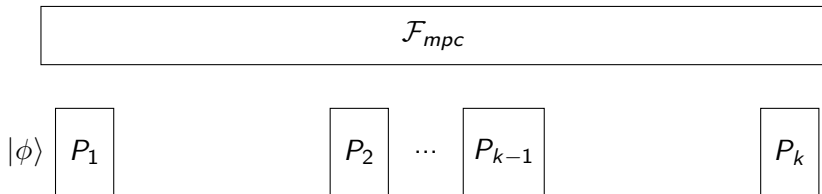
- P_{i^*} holds $C(|\psi\rangle|0^{2\lambda}\rangle)$
- All players (secret) share C
 - ▶ Players share random C_i 's s.t. $C_k \dots C_1 = C$
- Public authentication test
 - ▶ λ trap qubits used in the test
 - ▶ remaining λ to keep privacy/authentication even in the test

MPQC protocol - General idea

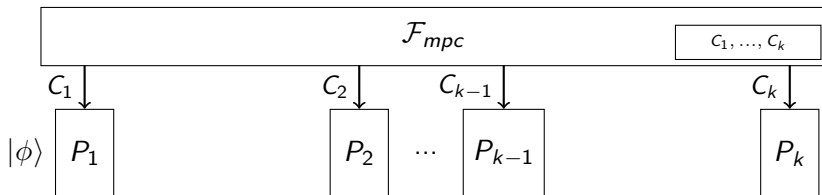
Focus on a single (pure) qubit

- P_{i^*} holds $C(|\psi\rangle|0^{2\lambda}\rangle)$
- All players (secret) share C
 - ▶ Players share random C_i 's s.t. $C_k \dots C_1 = C$
- Public authentication test
 - ▶ λ trap qubits used in the test
 - ▶ remaining λ to keep privacy/authentication even in the test
- Computation on encoded data

MPQC protocol - Encoding

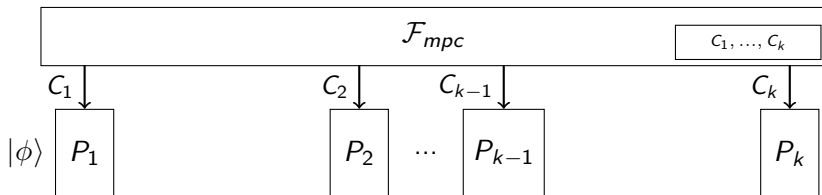


MPQC protocol - Encoding



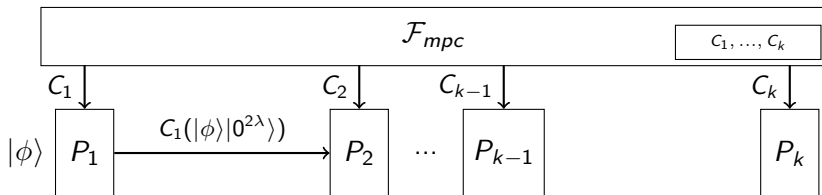
- $\mathcal{F}_{mpc}$ computes random $\{C_i\}_{2\lambda+1}$

MPQC protocol - Encoding



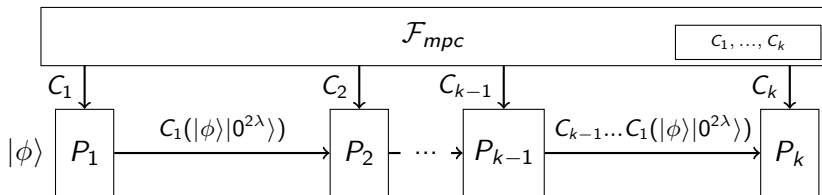
- $\mathcal{F}_{mpc}$ computes random $\{C_i\}_{2\lambda+1}$, parties apply C_i and send the state around the table

MPQC protocol - Encoding



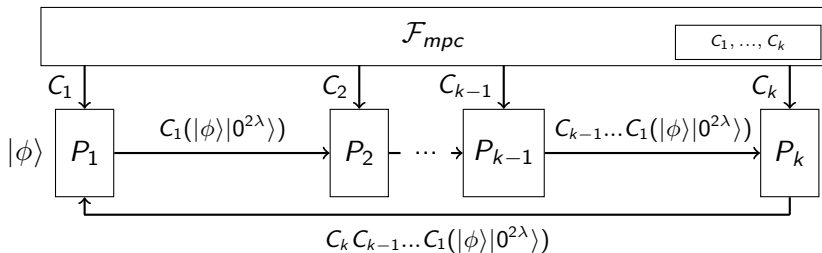
- $\mathcal{F}_{mpc}$ computes random $\{C_i\}_{2\lambda+1}$, parties apply C_i and send the state around the table

MPQC protocol - Encoding



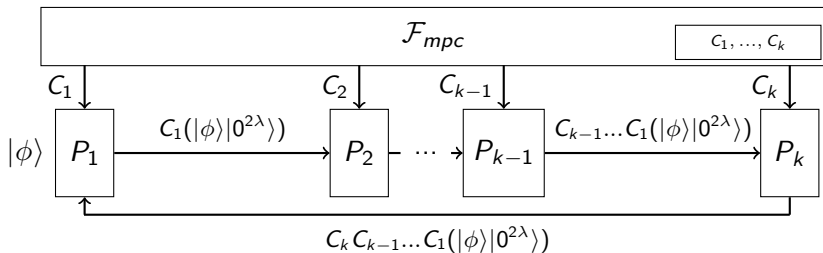
- $\mathcal{F}_{mpc}$ computes random $\{C_i\}_{2\lambda+1}$, parties apply C_i and send the state around the table

MPQC protocol - Encoding



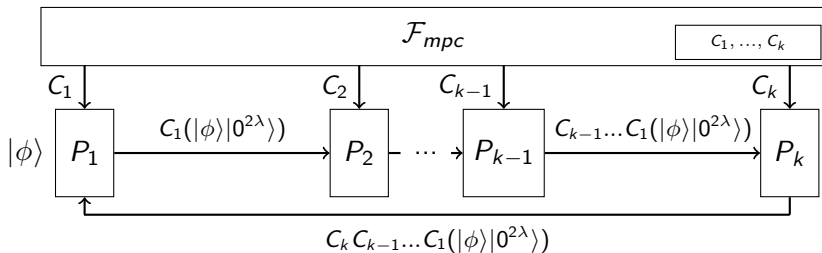
- $\mathcal{F}_{mpc}$ computes random $\{C_i\}_{2\lambda+1}$, parties apply C_i and send the state around the table

MPQC protocol - Encoding



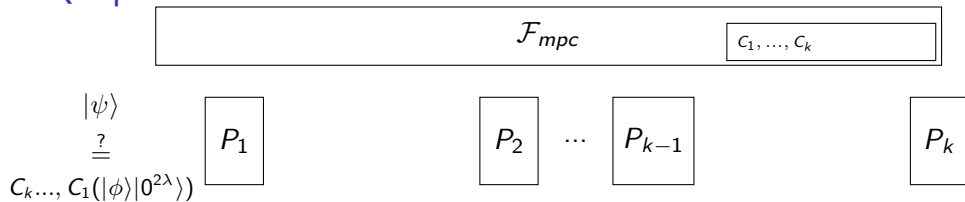
- $\mathcal{F}_{mpc}$ computes random $\{C_i\}_{2\lambda+1}$, parties apply C_i and send the state around the table
- How to prevent that any of the parties replaces the quantum state (or cheat arbitrarily)?

MPQC protocol - Encoding

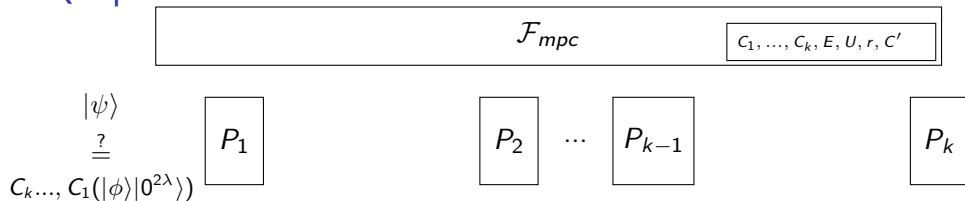


- $\mathcal{F}_{mpc}$ computes random $\{C_i\}_{2\lambda+1}$, parties apply C_i and send the state around the table
- How to prevent that any of the parties replaces the quantum state (or cheat arbitrarily)?
 - ▶ Public authentication test

MPQC protocol - Public authentication test



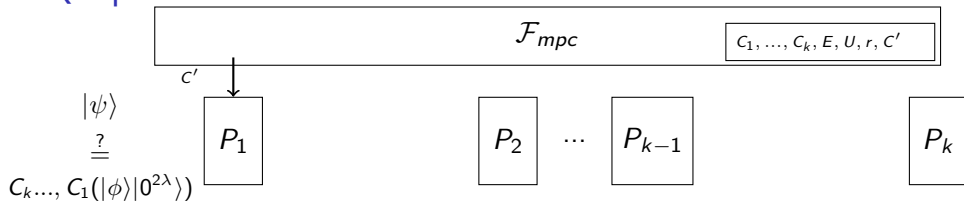
MPQC protocol - Public authentication test



- $\mathcal{F}_{mpc}$ computes random $C' \in \mathcal{C}_{2\lambda+1}$, $E \in \mathcal{C}_{\lambda+1}$, linear function U and $r \in \{0,1\}^\lambda$ s.t.

$$C' = (E \otimes X^r)(I_2 \otimes U)C_1^\dagger \dots C_{k-1}^\dagger C_k^\dagger$$

MPQC protocol - Public authentication test

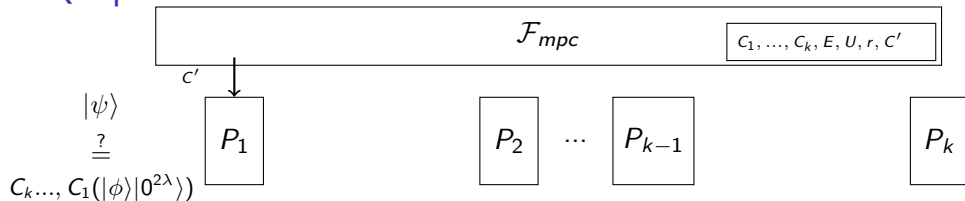


- $\mathcal{F}_{mpc}$ computes random $C' \in \mathcal{C}_{2\lambda+1}$, $E \in \mathcal{C}_{\lambda+1}$, linear function U and $r \in \{0,1\}^\lambda$ s.t.

$$C' = (E \otimes X^r)(I_2 \otimes U)C_1^\dagger \dots C_{k-1}^\dagger C_k^\dagger$$

- $\mathcal{F}_{MPC}$ sends only C' to P_1 and P_1 applies C' on $|\psi\rangle$

MPQC protocol - Public authentication test

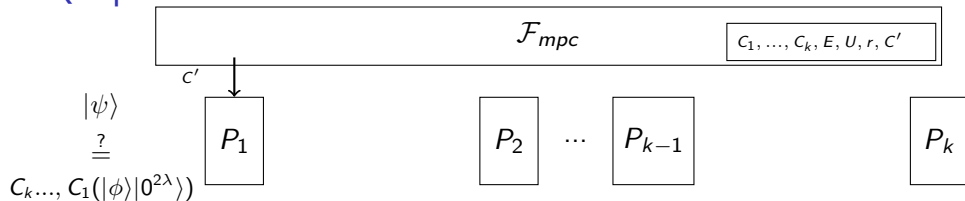


- $\mathcal{F}_{mpc}$ computes random $C' \in \mathcal{C}_{2\lambda+1}$, $E \in \mathcal{C}_{\lambda+1}$, linear function U and $r \in \{0,1\}^\lambda$ s.t.

$$C' = (E \otimes X^r)(I_2 \otimes U)C_1^\dagger \dots C_{k-1}^\dagger C_k^\dagger$$

- $\mathcal{F}_{MPC}$ sends only C' to P_1 and P_1 applies C' on $|\psi\rangle$
 - ▶ Honest case: $E(|\phi\rangle|0^\lambda\rangle)|r\rangle$

MPQC protocol - Public authentication test

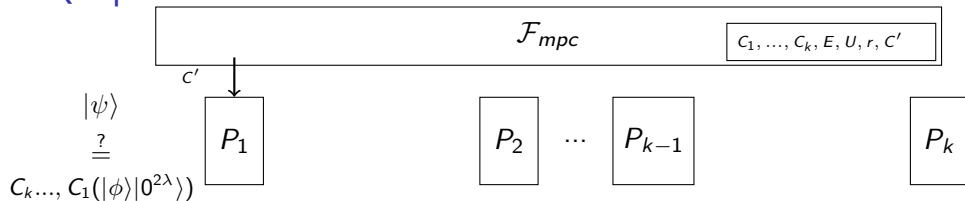


- $\mathcal{F}_{mpc}$ computes random $C' \in \mathcal{C}_{2\lambda+1}$, $E \in \mathcal{C}_{\lambda+1}$, linear function U and $r \in \{0,1\}^\lambda$ s.t.

$$C' = (E \otimes X^r)(I_2 \otimes U)C_1^\dagger \dots C_{k-1}^\dagger C_k^\dagger$$

- $\mathcal{F}_{MPC}$ sends only C' to P_1 and P_1 applies C' on $|\psi\rangle$
 - ▶ Honest case: $E(|\phi\rangle|0^\lambda\rangle)|r\rangle$
 - ▶ Dishonest case: last λ qubits are different of r with overwhelming probability

MPQC protocol - Public authentication test

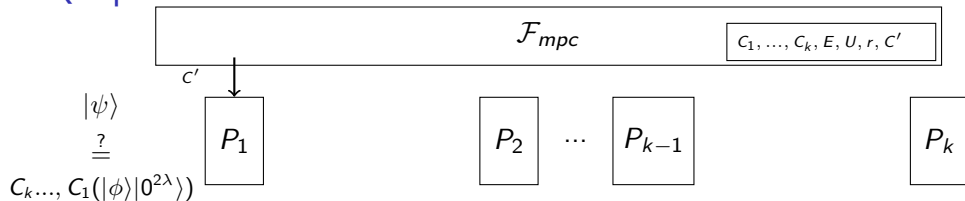


- $\mathcal{F}_{mpc}$ computes random $C' \in \mathcal{C}_{2\lambda+1}$, $E \in \mathcal{C}_{\lambda+1}$, linear function U and $r \in \{0,1\}^\lambda$ s.t.

$$C' = (E \otimes X^r)(I_2 \otimes U)C_1^\dagger \dots C_{k-1}^\dagger C_k^\dagger$$

- $\mathcal{F}_{MPC}$ sends only C' to P_1 and P_1 applies C' on $|\psi\rangle$
 - ▶ Honest case: $E(|\phi\rangle|0^\lambda\rangle)|r\rangle$
 - ▶ Dishonest case: last λ qubits are different of r with overwhelming probability
Unknown to all parties!

MPQC protocol - Public authentication test

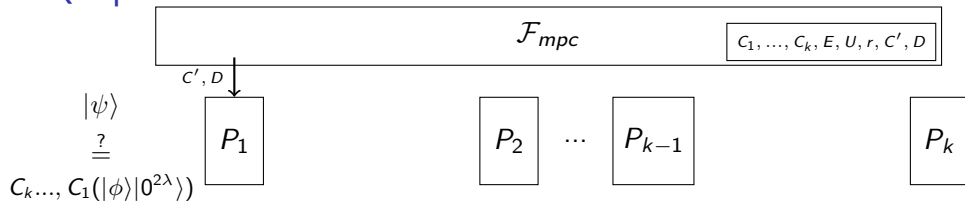


- $\mathcal{F}_{mpc}$ computes random $C' \in \mathcal{C}_{2\lambda+1}$, $E \in \mathcal{C}_{\lambda+1}$, linear function U and $r \in \{0,1\}^\lambda$ s.t.

$$C' = (E \otimes X^r)(I_2 \otimes U)C_1^\dagger \dots C_{k-1}^\dagger C_k^\dagger$$

- $\mathcal{F}_{MPC}$ sends only C' to P_1 and P_1 applies C' on $|\psi\rangle$
 - Honest case: $E(|\phi\rangle|0^\lambda\rangle)|r\rangle$
 - Dishonest case: last λ qubits are different of r with overwhelming probability
Unknown to all parties!
- Parties interact with $\mathcal{F}_{MPC}$ to check if the value of the traps is correct

MPQC protocol - Public authentication test

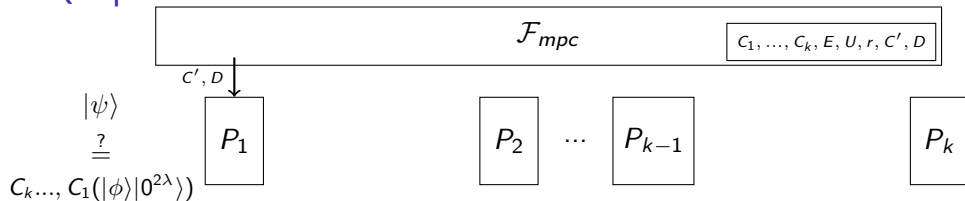


- $\mathcal{F}_{mpc}$ computes random $C' \in \mathcal{C}_{2\lambda+1}$, $E \in \mathcal{C}_{\lambda+1}$, linear function U and $r \in \{0,1\}^\lambda$ s.t.

$$C' = (E \otimes X^r)(I_2 \otimes U)C_1^\dagger \dots C_{k-1}^\dagger C_k^\dagger$$

- $\mathcal{F}_{MPC}$ sends only C' to P_1 and P_1 applies C' on $|\psi\rangle$
 - ▶ Honest case: $E(|\phi\rangle|0^\lambda\rangle)|r\rangle$
 - ▶ Dishonest case: last λ qubits are different of r with overwhelming probability
Unknown to all parties!
- Parties interact with $\mathcal{F}_{MPC}$ to check if the value of the traps is correct
- $\mathcal{F}_{MPC}$ sends new $D \in \mathcal{C}_{2\lambda+1}$ to P_1

MPQC protocol - Public authentication test



- $\mathcal{F}_{mpc}$ computes random $C' \in \mathcal{C}_{2\lambda+1}$, $E \in \mathcal{C}_{\lambda+1}$, linear function U and $r \in \{0,1\}^\lambda$ s.t.

$$C' = (E \otimes X^r)(I_2 \otimes U)C_1^\dagger \dots C_{k-1}^\dagger C_k^\dagger$$

- $\mathcal{F}_{MPC}$ sends only C' to P_1 and P_1 applies C' on $|\psi\rangle$
 - ▶ Honest case: $E(|\phi\rangle|0^\lambda\rangle)|r\rangle$
 - ▶ Dishonest case: last λ qubits are different of r with overwhelming probability

Unknown to all parties!

- Parties interact with $\mathcal{F}_{MPC}$ to check if the value of the traps is correct
- $\mathcal{F}_{MPC}$ sends new $D \in \mathcal{C}_{2\lambda+1}$ to P_1
- Similar procedure enables (secure) public measurement in the computational basis

MPQC protocol - Applying gates

- ① One-qubit Clifford D : can be performed by “changing the key”

$$C_k \dots C_1(|\phi\rangle|0^{2\lambda}\rangle) = C_k \dots C'_1(D|\phi\rangle|0^{2\lambda}\rangle), \text{ for } C'_1 = C_1 D^\dagger$$

MPQC protocol - Applying gates

- ① One-qubit Clifford D : can be performed by “changing the key”

$$C_k \dots C_1(|\phi\rangle|0^{2\lambda}\rangle) = C_k \dots C'_1(D|\phi\rangle|0^{2\lambda}\rangle), \text{ for } C'_1 = C_1 D^\dagger$$

- ② CNOT:

- ① Send two qubits to a single party (+ public authentication test)
- ② Re-encode the two qubits altogether (+ public authentication test)
- ③ Apply CNOT “changing the key”
- ④ Split the encoding of the two qubits (+ public authentication test)
- ⑤ Send each qubit to the corresponding party (+ public authentication test)

MPQC protocol - Applying gates

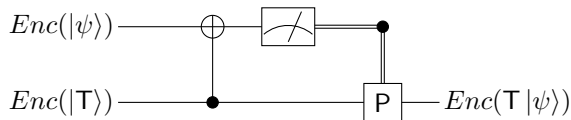
- 1 One-qubit Clifford D : can be performed by “changing the key”

$$C_k \dots C_1(|\phi\rangle|0^{2\lambda}\rangle) = C_k \dots C'_1(D|\phi\rangle|0^{2\lambda}\rangle), \text{ for } C'_1 = C_1 D^\dagger$$

- 2 CNOT:

- 1 Send two qubits to a single party (+ public authentication test)
- 2 Re-encode the two qubits altogether (+ public authentication test)
- 3 Apply CNOT “changing the key”
- 4 Split the encoding of the two qubits (+ public authentication test)
- 5 Send each qubit to the corresponding party (+ public authentication test)

- 3 T-gate:



MPQC protocol - creating T magic states

MPQC protocol - creating T magic states

- ① P_1 create $\text{poly}(\lambda, k)$ T -magic states
- ② Parties run sub-protocol to encode the (supposed) magic states
- ③ Each party tests a random subset
 - ▶ Locally decode (with the help of $\mathcal{F}_{mpc}$)
 - ▶ Check if the “raw” qubit is indeed $|T\rangle$
- ④ Use magic state distillation procedure to transform somewhat-good T -magic states into almost-perfect ones
 - ▶ Only need Clifford circuit + measurement

MPQC protocol - overall protocol

MPQC protocol - overall protocol

Protocol

- 1 Parties run sub-protocol to create $Enc(|T\rangle^{\otimes t})$
- 2 Parties run sub-protocol to encode each qubit
- 3 For each gate/measurement, parties run the corresponding sub-protocol
- 4 Each party decodes her own output (with the help of $\mathcal{F}_{MPC}$)