Sampling mixed quantum states

arXiv:1804.04730

Frédéric Dupuis *CNRS, LORIA, Nancy, France*

Joint work with Philippe Lamontagne, Serge Fehr and Louis Salvail

Sampling

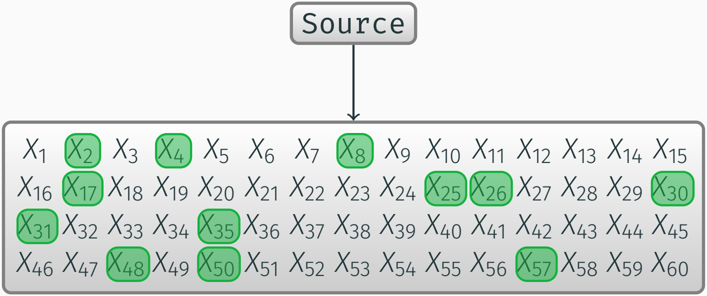
Classical certification

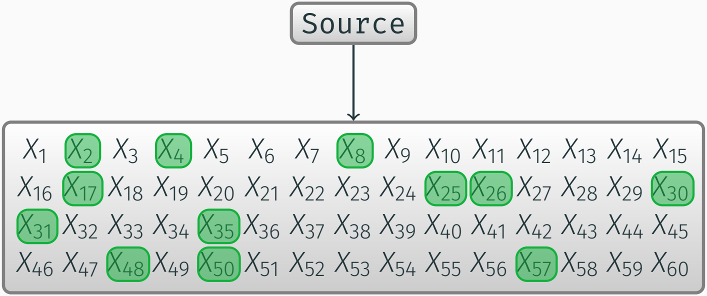
`Source`

$$\downarrow$$

$$\begin{array}{ccccccccccccccc}
X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 & X_8 & X_9 & X_{10} & X_{11} & X_{12} & X_{13} & X_{14} & X_{15} \\
X_{16} & X_{17} & X_{18} & X_{19} & X_{20} & X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} & X_{27} & X_{28} & X_{29} & X_{30} \\
X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} & X_{37} & X_{38} & X_{39} & X_{40} & X_{41} & X_{42} & X_{43} & X_{44} & X_{45} \\
X_{46} & X_{47} & X_{48} & X_{49} & X_{50} & X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} & X_{57} & X_{58} & X_{59} & X_{60}
\end{array}$$

Classical certification

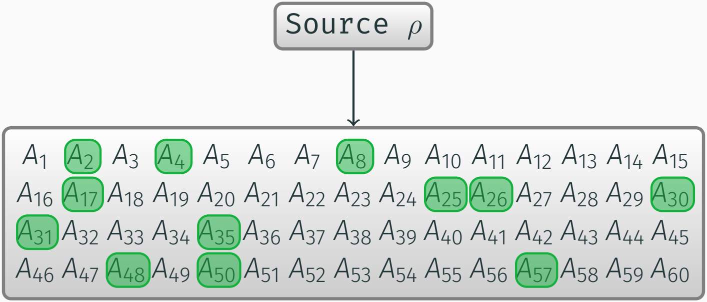




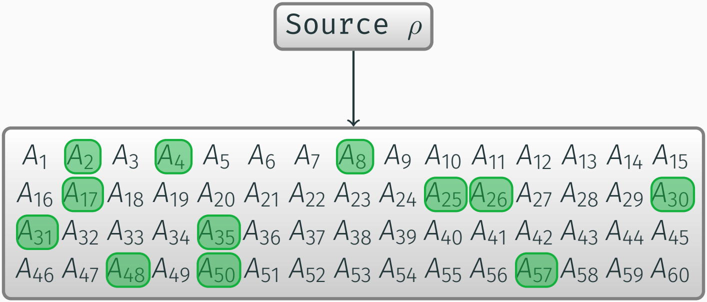
Suppose $X_i \in \{0, 1\}$. Then, sampling tells us:

- If we only see zeros in the sample $\overset{whp}{\Rightarrow}$ we should have at most δn 1's in the rest

Quantum certification



- Now, each A_i is a qubit
- Suppose we measure all the qubits in the sample in the computational basis, get all zeros
- What can we say about the state?



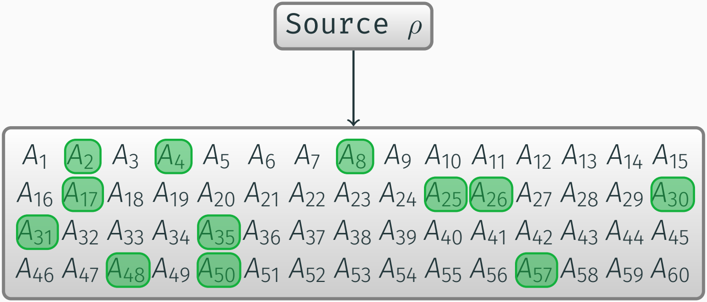
- We can define a *low-error subspace*

$$\mathcal{T}_\varepsilon := \mathsf{span}\left\{\left|x_1^{n-k}\right\rangle : x_1^{n-k} \text{ has at most } \varepsilon n \text{ 1's}\right\}$$

- Statement:

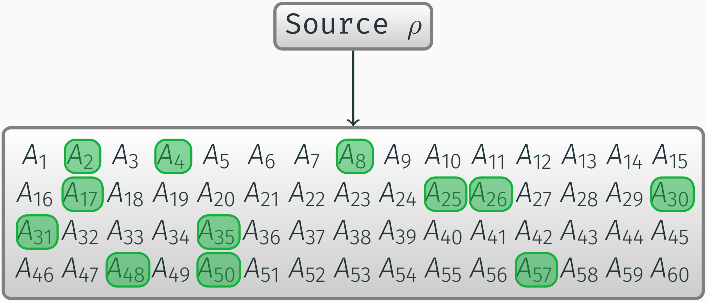
$$\mathsf{tr}\left[\rho_{\square}\Pi_{\mathcal{T}_\varepsilon}\right] \geqslant 1 - \mathsf{negl}$$

Quantum sampling



- Bouman and Fehr showed that any classical sampling procedure has a quantum analogue
- This works as long as we're certifying *pure* states
- What happens if we want to certify mixed states?

Certifying mixed states



- We now want to certify that most positions are in the mixed state φ
- We could measure sampled positions in the diagonal basis of φ, see if we get the right statistics
- This fails: a pure state with the right stats would pass the test

Certifying mixed states

- The task is impossible as it stands:
  - $\varphi^{\otimes n}$ is a mixture of pure states, each of which should fail the test
- Classically, the task also makes no sense
  - Looking at a bitstring, what probability distribution did it come from?
- It makes sense if we can ask for purifications:

$$\varphi_A \to |\varphi\rangle_{AR}$$

A mixed state certification protocol

A *interactive game* between two players: a Prover and a Verifier

A mixed state certification protocol

A *interactive game* between two players: a Prover and a Verifier

Goal

Verifier wants to certify that his state is close to $\varphi^{\otimes n}$. Prover wants to fool the verifier into thinking he has the right state even though it's not the case.

A mixed state certification protocol

A *interactive game* between two players: a Prover and a Verifier

Goal

Verifier wants to certify that his state is close to $\varphi^{\otimes n}$. Prover wants to fool the verifier into thinking he has the right state even though it's not the case.

- P. Prepare $|\varphi\rangle_{AR}^{\otimes n}$, send A^n to verifier.
- V. Choose a random sample, announce it to prover.
- P. Send R for each position in sample.
- V. Measure $\{|\varphi\rangle\langle\varphi|_{AR}, \mathbb{I} - |\varphi\rangle\langle\varphi|_{AR}\}$ for each joint system AR in sample.
- V. Accept if no errors, reject otherwise.

Is this protocol secure? What does it mean to be secure?

Applications

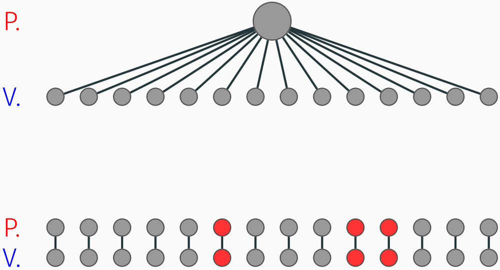
This has some applications in cryptography:

- Coin tossing: Alice prepares n EPR pairs, Bob certifies them, then they measure in the computational basis.
  - Caveat: we still get a few errors, no way to get rid of them $\Rightarrow$ we get a source of min-entropy arbitrarily close to n
- Preparing "magic states" for multiparty computation protocols

Defining security

How do we define security? Tempting definition:

- With high probability, the prover could produce purifications of the remaining systems with at most εn errors

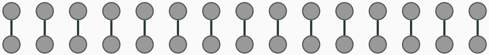


This definition doesn't work, because of *postselection attacks*

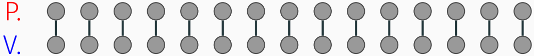
Postselection attack

P.

V.

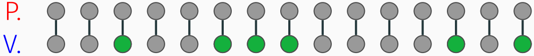


Postselection attack



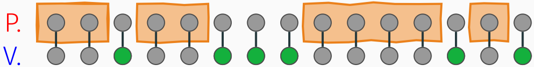
1. Learns sample

Postselection attack



1. Learns sample
2. Measures qubits

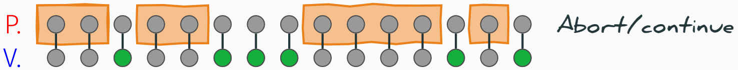
Postselection attack



1. Learns sample
2. Measures qubits
3. Aborts based on result

Postselection attacks

Postselection attack

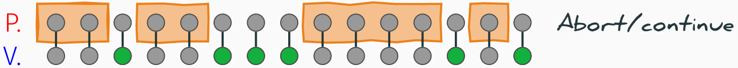


1. Learns sample
2. Measures qubits
3. Aborts based on result

Post-selection

Postselection attacks

Postselection attack



1. Learns sample
2. Measures qubits
3. Aborts based on result

Post-selection

Example

Prepare $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)^{\otimes n}$, measure positions outside of sample, abort if result $\neq |0\rangle^{\otimes n-k}$.

Resulting state always $|0\rangle^{\otimes n-k}$

What *can* the prover do?

What *can* the prover do?

An “undetectable” attack

The prover can

- prepare the honest state, up to a few errors,
- prepare a mixture/superposition of such states,
- purify this mixture, and
- post-select on a measurement outcome.

What *can* the prover do?

An "undetectable" attack

The prover can

- prepare the honest state, up to a few errors,
- prepare a mixture/superposition of such states,
- purify this mixture, and
- post-select on a measurement outcome.

$$|\varphi\rangle^{\otimes n} = |\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle$$

What *can* the prover do?

An “undetectable” attack

The prover can

- prepare the honest state, up to a few errors,
- prepare a mixture/superposition of such states,
- purify this mixture, and
- post-select on a measurement outcome.

$$|\varphi\rangle^{\otimes n} = |\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle$$

$$|\psi_e\rangle = |\varphi\rangle{\color{orange}|\varphi\rangle}|\varphi\rangle|\varphi\rangle|\varphi\rangle{\color{orange}|\varphi\rangle}|\varphi\rangle{\color{orange}|\varphi\rangle}|\varphi\rangle|\varphi\rangle|\varphi\rangle{\color{orange}|\varphi\rangle}|\varphi\rangle|\varphi\rangle{\color{orange}|\varphi\rangle}|\varphi\rangle|\varphi\rangle$$

An "undetectable" attack

The prover can

- prepare the honest state, up to a few errors,
- prepare a mixture/superposition of such states,
- purify this mixture, and
- post-select on a measurement outcome.

$$|\varphi\rangle^{\otimes n} = |\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle|\varphi\rangle$$

$$|\psi_e\rangle = |\varphi\rangle\rotatebox{-30}{$|\varphi\rangle$}|\varphi\rangle|\varphi\rangle|\varphi\rangle\rotatebox{-40}{$|\varphi\rangle$}|\varphi\rangle\rotatebox{90}{$|\varphi\rangle$}|\varphi\rangle|\varphi\rangle|\varphi\rangle\rotatebox{-40}{$|\varphi\rangle$}|\varphi\rangle|\varphi\rangle\rotatebox{60}{$|\varphi\rangle$}|\varphi\rangle|\varphi\rangle$$

$$\rho_{A^n R^n} = \sum_e p_e |\psi_e\rangle\langle\psi_e|$$

What *can* the prover do?

An “undetectable” attack

The prover can

- prepare the honest state, up to a few errors,
- prepare a mixture/superposition of such states,
- purify this mixture, and
- post-select on a measurement outcome.

$$\cdots$$

$$|\psi_e\rangle = |\varphi\rangle \cancel{|\varphi\rangle} |\varphi\rangle|\varphi\rangle|\varphi\rangle \cancel{|\varphi\rangle} |\varphi\rangle \cancel{|\varphi\rangle} |\varphi\rangle|\varphi\rangle|\varphi\rangle \cancel{|\varphi\rangle} |\varphi\rangle|\varphi\rangle \cancel{|\varphi\rangle} |\varphi\rangle|\varphi\rangle$$

$$\rho_{A^n R^n} = \sum_e p_e |\psi_e\rangle\langle\psi_e|$$

$$|\Psi\rangle_{A^n R^n E} = \textstyle\sum_e \sqrt{p_e} |\psi_e\rangle_{A^n R^n} \otimes |\tau_e\rangle_E$$

What *can* the prover do?

An "undetectable" attack

The prover can

- prepare the honest state, up to a few errors,
- prepare a mixture/superposition of such states,
- purify this mixture, and
- post-select on a measurement outcome.

$$\rho_{A^n R^n} = \sum_{e}^{\cdots} p_e |\psi_e\rangle\langle\psi_e|$$

$$|\Psi\rangle_{A^n R^n E} = \sum_e \sqrt{p_e} |\psi_e\rangle_{A^n R^n} \otimes |\tau_e\rangle_E$$

$$|\hat{\Psi}\rangle_{A^n R^n E} = \mathbb{I}_{A^n} \otimes M_{R^n E} |\Psi\rangle_{A^n R^n E}$$

What *can* the prover do?

An "undetectable" attack

The prover can

- prepare the honest state, up to a few errors,
- prepare a mixture/superposition of such states,
- purify this mixture, and
- post-select on a measurement outcome.

$$\rho_{A^n R^n} = \sum_{e}^{\cdots} p_e |\psi_e\rangle\langle\psi_e|$$

$$\boxed{|\Psi\rangle_{A^n R^n E} = \textstyle\sum_e \sqrt{p_e}|\psi_e\rangle_{A^n R^n} \otimes |\tau_e\rangle_E}$$

ideal state

$$|\hat{\Psi}\rangle_{A^n R^n E} = \mathbb{I}_{A^n} \otimes M_{R^n E}|\Psi\rangle_{A^n R^n E}$$

Defining success

Definition (Soundness)

For any strategy for the prover, the output state ρ_{A^n} of the verifier is s.t.

RHS is "rough approximation" of LHS

$$\rho_{A^n} \leqslant p_n \cdot \psi_{A^n} + \sigma$$

where p_n is polynomial in n, ψ_{A^n} is part of an ideal state $|\psi\rangle_{A^n R^n E}$ and $\mathrm{tr}(\sigma) \leq \mathrm{negl}(n)$.

Defining success

Definition (Soundness)

For any strategy for the prover, the output state ρ_{A^n} of the verifier is s.t.

RHS is "rough approximation" of LHS

$$\rho_{A^n} \leqslant p_n \cdot \psi_{A^n} + \sigma$$

where p_n is polynomial in n, ψ_{A^n} is part of an ideal state $|\psi\rangle_{A^n R^n E}$ and $\mathbf{tr}(\sigma) \leq \text{negl}(n)$.

Application

For any "bad event",

$$\Pr[\text{bad event} \mid \rho_{A^n}] \leq p_n \Pr[\text{bad event} \mid \psi_{A^n}] + \text{negl}(n)$$

Secure application if $\Pr[\text{bad event} \mid \psi_{A^n}]$ is negligible.

Main result

Our sampling protocol:

P. Prepare $|\varphi\rangle_{AR}^{\otimes n}$, send A^n to verifier.

V. Choose a random sample, announce it to prover.

P. Send R for each position in sample.

V. Measure $\{|\varphi\rangle\langle\varphi|_{AR}, \mathbb{I} - |\varphi\rangle\langle\varphi|_{AR}\}$ for each joint system AR in sample.

V. Accept if no errors, reject otherwise.

Theorem (Main)

This protocol is sound.

Proof Tools and Sketch

Permutations and sampling are closely related

Permutations and sampling are closely related

Choosing a random subset of size k of a population

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22

Permutations and sampling are closely related

Choosing a random subset of size k of a population

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22

is the same as randomly permuting the population and picking the first k elements

1, 22, 20, 0, 11, 12, 14, 8, 9, 3, 18, 15, 6, 2, 17, 5, 19, 10, 13, 4, 21, 16, 7

Permutations and sampling are closely related

Choosing a random subset of size k of a population

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22

is the same as randomly permuting the population and picking the first k elements

1, 22, 20, 0, 11, 12, 14, 8, 9, 3, 18, 15, 6, 2, 17, 5, 19, 10, 13, 4, 21, 16, 7

Sampling is "invariant under permutation".

Permutation invariance is a powerful tool

- Permutations are bijections of the form

$$\pi : \{1, \ldots, n\} \to \{1, \ldots, n\} \qquad \text{e.g. } \pi = \begin{pmatrix} 12345 \\ 53421 \end{pmatrix}$$

Permutation invariance is a powerful tool

- Permutations are bijections of the form

$$\pi : \{1, \ldots, n\} \to \{1, \ldots, n\} \qquad \text{e.g. } \pi = \begin{pmatrix} 12345 \\ 53421 \end{pmatrix}$$

- Action on $\mathcal{H}^{\otimes n}$ is

$$\pi |\psi_1\rangle|\psi_2\rangle|\psi_3\rangle|\psi_4\rangle|\psi_5\rangle = |\psi_5\rangle|\psi_4\rangle|\psi_2\rangle|\psi_3\rangle|\psi_1\rangle$$

Permutation invariance is a powerful tool

- Permutations are bijections of the form

$$\pi : \{1, \dots, n\} \to \{1, \dots, n\} \qquad \text{e.g. } \pi = \begin{pmatrix} 12345 \\ 53421 \end{pmatrix}$$

- Action on $\mathcal{H}^{\otimes n}$ is

$$\pi |\psi_1\rangle |\psi_2\rangle |\psi_3\rangle |\psi_4\rangle |\psi_5\rangle = |\psi_5\rangle |\psi_4\rangle |\psi_2\rangle |\psi_3\rangle |\psi_1\rangle$$

- Permutation invariant operators are approximated by mixtures of i.i.d. operators [CKR09]

$$\rho = \pi \rho \pi^* \quad \forall \pi \implies \rho \leq p(n) \int \theta^{\otimes n} d\theta$$

Permutation invariance is a powerful tool

- Permutations are bijections of the form

$$\pi : \{1, \ldots, n\} \to \{1, \ldots, n\} \qquad \text{e.g. } \pi = \begin{pmatrix} 12345 \\ 53421 \end{pmatrix}$$

- Action on $\mathcal{H}^{\otimes n}$ is

$$\pi |\psi_1\rangle |\psi_2\rangle |\psi_3\rangle |\psi_4\rangle |\psi_5\rangle = |\psi_5\rangle |\psi_4\rangle |\psi_2\rangle |\psi_3\rangle |\psi_1\rangle$$

- Permutation invariant operators are approximated by mixtures of i.i.d. operators [CKR09]

$$\rho = \pi \rho \pi^* \quad \forall \pi \implies \rho \leq p(n) \int \theta^{\otimes n} d\theta$$

Permutation invariance is a powerful tool

- Permutations are bijections of the form

$$\pi : \{1, \ldots, n\} \to \{1, \ldots, n\} \qquad \text{e.g. } \pi = \begin{pmatrix} 12345 \\ 53421 \end{pmatrix}$$

- Action on $\mathcal{H}^{\otimes n}$ is

$$\pi|\psi_1\rangle|\psi_2\rangle|\psi_3\rangle|\psi_4\rangle|\psi_5\rangle = |\psi_5\rangle|\psi_4\rangle|\psi_2\rangle|\psi_3\rangle|\psi_1\rangle$$

- Permutation invariant operators are approximated by mixtures of i.i.d. operators [CKR09]

$$\rho = \pi\rho\pi^* \quad \forall\pi \implies \rho \leq p(n) \int \theta^{\otimes n} d\theta$$

Permutation invariance is a powerful tool

- Permutations are bijections of the form

$$\pi : \{1, \ldots, n\} \to \{1, \ldots, n\} \qquad \text{e.g. } \pi = \begin{pmatrix} 12345 \\ 53421 \end{pmatrix}$$

- Action on $\mathcal{H}^{\otimes n}$ is

$$\pi|\psi_1\rangle|\psi_2\rangle|\psi_3\rangle|\psi_4\rangle|\psi_5\rangle = |\psi_5\rangle|\psi_4\rangle|\psi_2\rangle|\psi_3\rangle|\psi_1\rangle$$

- Permutation invariant operators are approximated by mixtures of i.i.d. operators [CKR09]

$$\rho = \pi\rho\pi^* \quad \forall\pi \implies \rho \leq p(n) \int \theta^{\otimes n} d\theta$$

Permutation invariance is a powerful tool

- Permutations are bijections of the form

$$\pi : \{1, \ldots, n\} \to \{1, \ldots, n\} \qquad \text{e.g. } \pi = \begin{pmatrix} 12345 \\ 53421 \end{pmatrix}$$

- Action on $\mathcal{H}^{\otimes n}$ is

$$\pi|\psi_1\rangle|\psi_2\rangle|\psi_3\rangle|\psi_4\rangle|\psi_5\rangle = |\psi_5\rangle|\psi_4\rangle|\psi_2\rangle|\psi_3\rangle|\psi_1\rangle$$

- Permutation invariant operators are approximated by mixtures of i.i.d. operators [CKR09]

$$\rho = \pi\rho\pi^* \quad \forall\pi \implies \rho \leq p(n) \int \theta^{\otimes n} d\theta$$

Permutation invariance is a powerful tool

- Permutations are bijections of the form

$$\pi : \{1, \ldots, n\} \to \{1, \ldots, n\} \qquad \text{e.g. } \pi = \binom{12345}{53421}$$

- Action on $\mathcal{H}^{\otimes n}$ is

$$\pi |\psi_1\rangle|\psi_2\rangle|\psi_3\rangle|\psi_4\rangle|\psi_5\rangle = |\psi_5\rangle|\psi_4\rangle|\psi_2\rangle|\psi_3\rangle|\psi_1\rangle$$

- Permutation invariant operators are approximated by mixtures of i.i.d. operators [CKR09]

$$\rho = \pi\rho\pi^* \quad \forall\pi \implies \rho \leq p(n) \int \theta^{\otimes n} d\theta = \mathbb{E}_\Theta[\Theta^{\otimes n}]$$

Rough sketch of the proof

- Protocol is invariant under permutation of verifier's registers **if** prover knows π.
- Randomly permute A^n, give π to prover.
- No loss of generality in assuming prover purifies choice of π.
- Equivalent to attack using permutation invariant $\rho_{A^n R^n}$
- $\Pr[\text{Attack} \mid \rho_{A^n R^n}] \leq \Pr[\text{Attack} \mid \int \theta_{AR}^{\otimes n}\, d\theta] \leq \text{negl}(n)$
- Still need to unpermute verifier's output

Rough sketch of the proof

- Protocol is invariant under permutation of verifier's registers **if** prover knows π.
- Randomly permute A^n, give π to prover.
- No loss of generality in assuming prover purifies choice of π.
- Equivalent to attack using permutation invariant $\rho_{A^n R^n}$
- $\Pr[\text{Attack} \mid \rho_{A^n R^n}] \leq \Pr[\text{Attack} \mid \int \theta_{AR}^{\otimes n}\, d\theta] \leq \text{negl}(n)$
- Still need to unpermute verifier's output

Rough sketch of the proof

- Protocol is invariant under permutation of verifier's registers **if** prover knows π.
- Randomly permute A^n, give π to prover.
- No loss of generality in assuming prover purifies choice of π.
- Equivalent to attack using permutation invariant $\rho_{A^n R^n}$
- $\Pr[\text{Attack} \mid \rho_{A^n R^n}] \leq \Pr[\text{Attack} \mid \int \theta_{AR}^{\otimes n}\, d\theta] \leq \text{negl}(n)$
- Still need to unpermute verifier's output

Rough sketch of the proof

- Protocol is invariant under permutation of verifier's registers **if** prover knows π.
- Randomly permute A^n, give π to prover.
- No loss of generality in assuming prover purifies choice of π.
- Equivalent to attack using permutation invariant $\rho_{A^n R^n}$
- $\Pr[\text{Attack} \mid \rho_{A^n R^n}] \leq \Pr[\text{Attack} \mid \int \theta_{AR}^{\otimes n} \, d\theta] \leq \mathrm{negl}(n)$
- Still need to unpermute verifier's output

Rough sketch of the proof

- Protocol is invariant under permutation of verifier's registers **if** prover knows π.
- Randomly permute A^n, give π to prover.
- No loss of generality in assuming prover purifies choice of π.
- Equivalent to attack using permutation invariant $\rho_{A^n R^n}$
- $\Pr[\text{Attack} \mid \rho_{A^n R^n}] \leq \Pr[\text{Attack} \mid \int \theta_{AR}^{\otimes n}\, d\theta] \leq \mathrm{negl}(n)$
- Still need to unpermute verifier's output

Rough sketch of the proof

- Protocol is invariant under permutation of verifier's registers **if** prover knows π.
- Randomly permute A^n, give π to prover.
- No loss of generality in assuming prover purifies choice of π.
- Equivalent to attack using permutation invariant $\rho_{A^n R^n}$
- $\Pr[\text{Attack} \mid \rho_{A^n R^n}] \leq \Pr[\text{Attack} \mid \int \theta_{AR}^{\otimes n}\, d\theta] \leq \text{negl}(n)$
- Still need to unpermute verifier's output

Rough sketch of the proof

- Protocol is invariant under permutation of verifier's registers **if** prover knows π.
- Randomly permute A^n, give π to prover.
- No loss of generality in assuming prover purifies choice of π.
- Equivalent to attack using permutation invariant $\rho_{A^n R^n}$
- $\Pr[\text{Attack} \mid \rho_{A^n R^n}] \leq \Pr[\text{Attack} \mid \int \theta_{AR}^{\otimes n}\, d\theta] \leq \text{negl}(n)$
  easy
- Still need to unpermute verifier's output hard

Conclusion

Conclusion and open problems

Conclusion

- Certifying mixed states is possible **if** you have access to the source.
- Suitable for use in a cryptographic setting.
- Permutation invariance plays essential role in proof.

Conclusion and open problems

Conclusion

- Certifying mixed states is possible **if** you have access to the source.
- Suitable for use in a cryptographic setting.
- Permutation invariance plays essential role in proof.

Conclusion and open problems

Conclusion

- Certifying mixed states is possible **if** you have access to the source.
- Suitable for use in a cryptographic setting.
- Permutation invariance plays essential role in proof.

Conclusion and open problems

Conclusion

- Certifying mixed states is possible **if** you have access to the source.
- Suitable for use in a cryptographic setting.
- Permutation invariance plays essential role in proof.

Open problems

- Certifying arbitrary reference states (vs $\varphi^{\otimes n}$)
- Do sampling in the more general sense of estimating the error rate.

Conclusion and open problems

Conclusion

- Certifying mixed states is possible **if** you have access to the source.
- Suitable for use in a cryptographic setting.
- Permutation invariance plays essential role in proof.

Open problems

- Certifying arbitrary reference states (vs $\varphi^{\otimes n}$)
- Do sampling in the more general sense of estimating the error rate.

Conclusion and open problems

Conclusion

- Certifying mixed states is possible **if** you have access to the source.
- Suitable for use in a cryptographic setting.
- Permutation invariance plays essential role in proof.

Open problems

- Certifying arbitrary reference states (vs $\varphi^{\otimes n}$)
- Do sampling in the more general sense of estimating the error rate.

Thank you!

The paper: `arXiv:1804.04730`