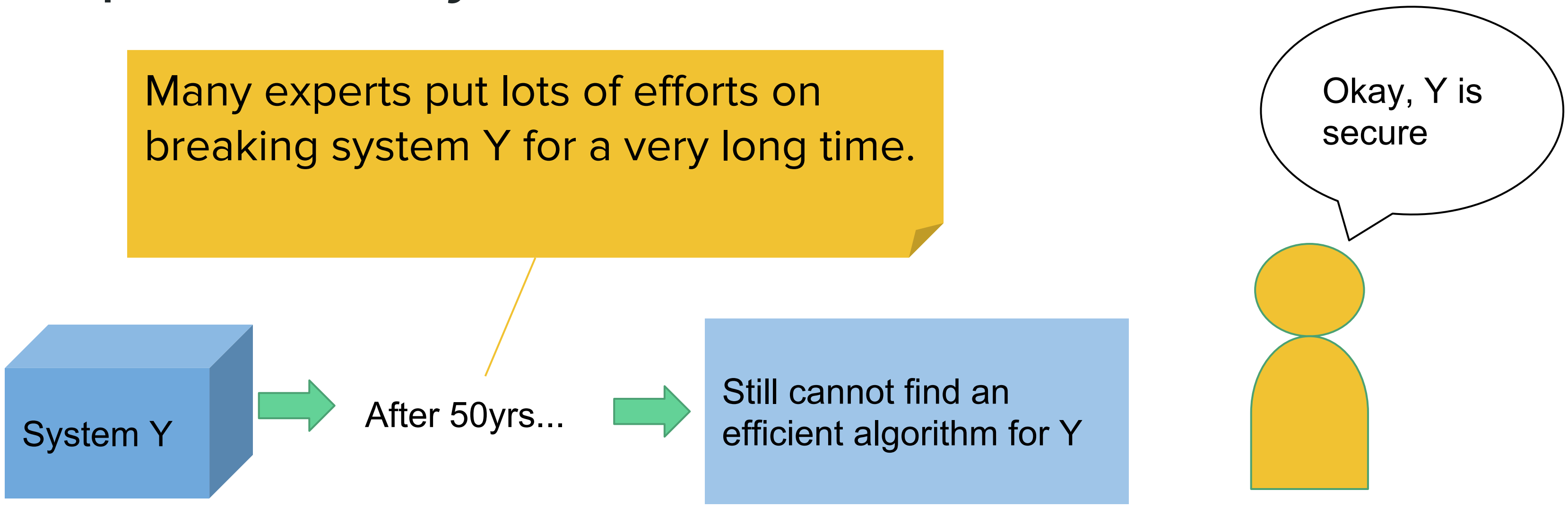
On basing one-way permutations on NP-hard problems under quantum reductions

Nai-Hui Chia
(PennState to UTAustin)

Joint work with Sean Hallgren (PennState)
and Fang Song (PortlandState to TAMU)

How do people say a crypto system is computationally secure?



How do people say a crypto system is computationally secure?

Many experts put lots of efforts on breaking system Y for a very long time.

Okay, Y is secure

System Y

After 50yrs...

Still cannot find an efficient algorithm for Y

Do we really need to wait 50yrs?

How do people say a crypto system is computationally secure?

Many experts put lots of efforts on breaking system Y for a very long time.

Okay, Y is secure

System Y

After 50yrs...

Still cannot find an efficient algorithm for Y

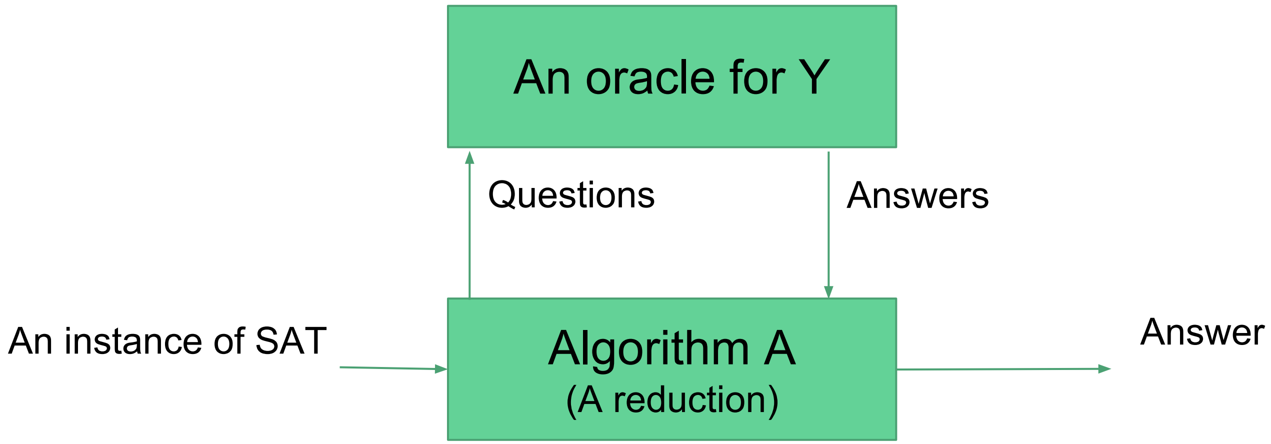
Do we really need to wait 50yrs?

SAT

- SAT has already been studied for >50yrs.
- SAT is hard (NP-complete)
- P≠NP (people believe)

Use SAT to show Problem Y is hard.

Show Y is hard by a reduction from SAT: SAT ≤ Y



SAT ≤ Y:

- An efficient algorithm *A* solving SAT by using an oracle for Y.
- Algorithm *A* and (Questions, Answers) can be either classical or quantum!

SAT ≤ Y ⇒ No efficient algorithm can break system Y unless NP = P.

Consider Y as inverting one-way functions

- Functions which are easy to compute but hard to invert.
- A fundamental cryptographic primitive. The existence of one-way functions implies
  - Pseudorandom generators
  - Digital signature scheme
  - Message Authentication Codes
  - ....

Consider Y as inverting one-way functions

- Functions which are easy to compute but hard to invert.
- A fundamental cryptographic primitive. The existence of one-way functions implies
  - Pseudorandom generators
  - Digital signature scheme
  - Message Authentication Codes
  - ....

Can inverting one-way functions be as hard as SAT?

One-way functions

- Functions which are easy to compute but hard to invert.
- A fundamental cryptographic primitive. It implies
  - Pseudorandom generators
  - Digital signature scheme
  - Message Authentication Codes
  - ....

Can inverting one-way functions be as hard as SAT?

- **SAT $\leq_c$ Inverting a one-way permutation ⇒ PH collapses [Brassard96].**
- **SAT $\leq_c$ Inverting a one-way function ⇒ PH collapses,**
  - **when the reductions are non-adaptive [AGGM05] or the functions are preimage verifiable[AGGM05,BB15].**

One-way functions

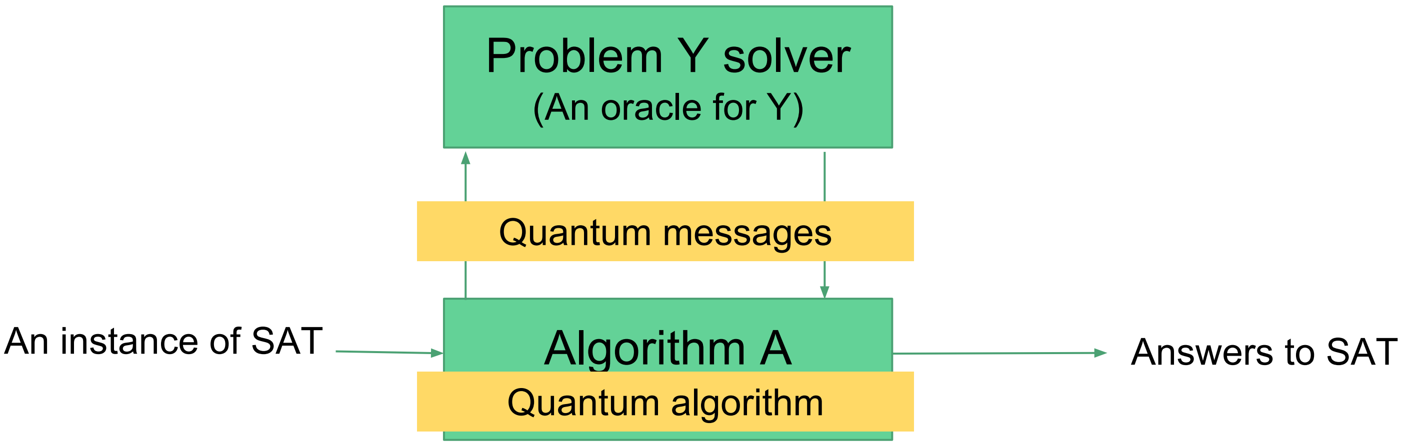
- Functions which are easy to compute but hard to invert.
- A fundamental cryptographic primitive. It implies
  - Pseudorandom generators
  - Digital signature scheme
  - Message Authentication Codes
  - ....

Can inverting one-way functions be as hard as SAT?

- **SAT $\leq_c$ Inverting a one-way permutation ⇒ PH collapses [Brassard96].**
- **SAT $\leq_c$ Inverting a one-way function ⇒ PH collapses,**
  - **when the reductions are non-adaptive[AGGM05] or the functions are preimage verifiable[AGGM05, BB15].**

Only classical reductions are considered!

We are interested in quantum reductions



Hard problems
(e.g., NP-hard problems)

$\leq_{\text{quantum}}$

Computational tasks
(e.g., inverting one-way functions)

Do these reductions exist?

- **SAT $\leq_c$ Inverting a one-way permutation ⇒ coNP ⊆ AM ⇒ PH collapses [Brassard96].**
- **SAT $\leq_c$ Inverting a one-way function ⇒ PH collapses,**
  - **when the reductions are non-adaptive[BT06] or the functions are preimage verifiable[].**

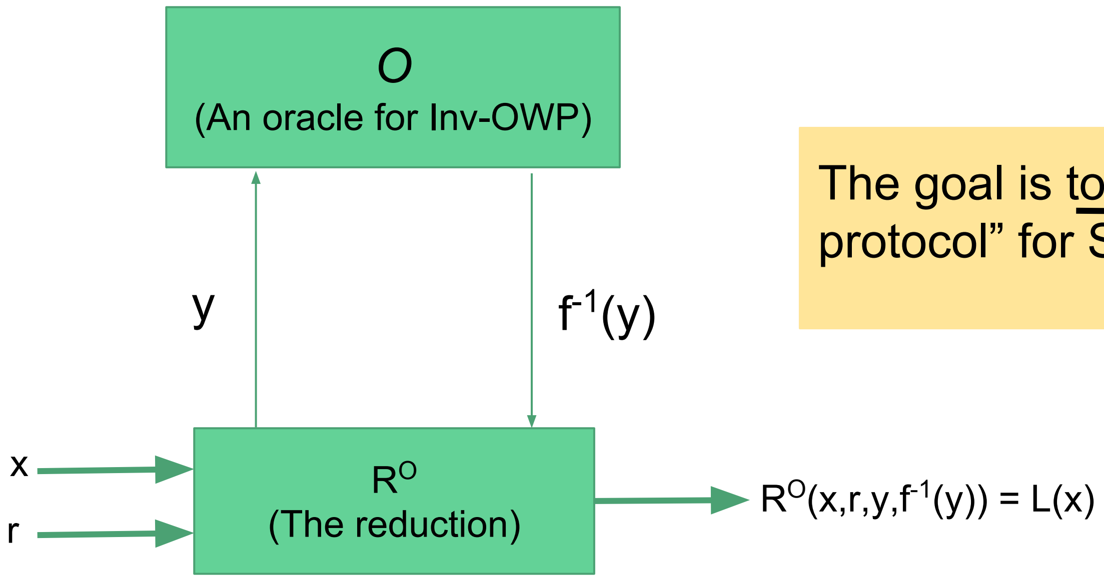
Our results

SAT $\leq_q$ Inverting a one-way permutation (Inv-OWP) ⇒ coNP ⊆ QIP(2), where

- our result has the restrictions that the reductions are non-adaptive and the distribution of the questions to the oracle are not far from the uniform distribution.
- It is not known if **coNP ⊆ QIP(2).**

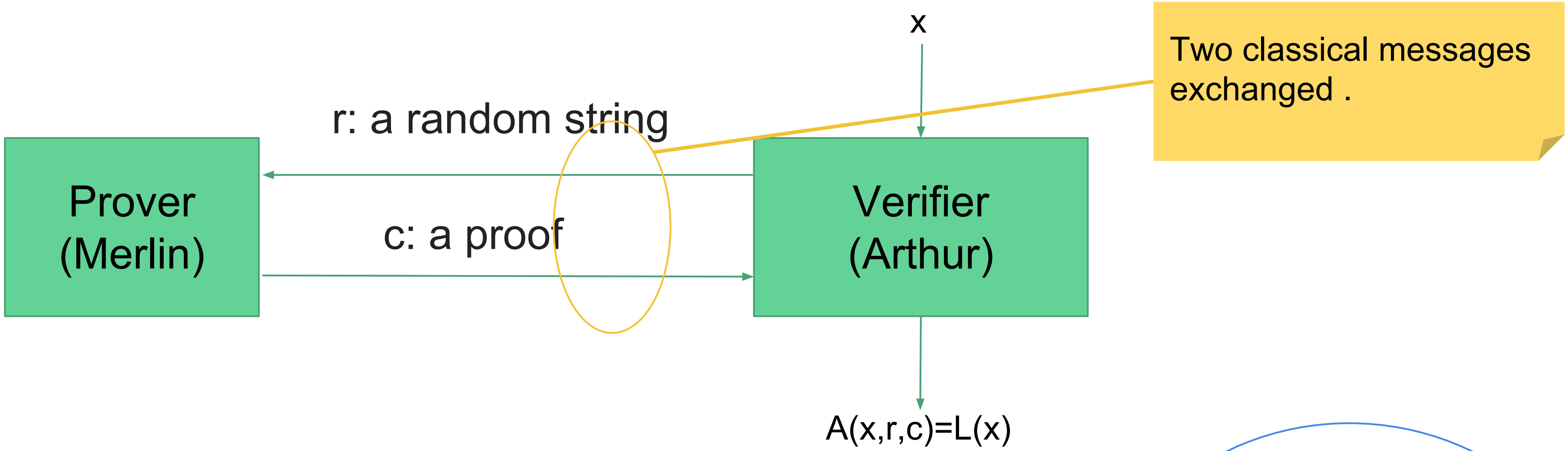
NP-hard Problems $\leq_c$ Inv-OWP $\Rightarrow$ coNP $\subseteq$ AM

Theorem [Brassad96]: SAT $\leq_c$ Inv-OWP $\Rightarrow$ coNP $\subseteq$ AM $\Rightarrow$ The polynomial hierarchy collapses to the second level.



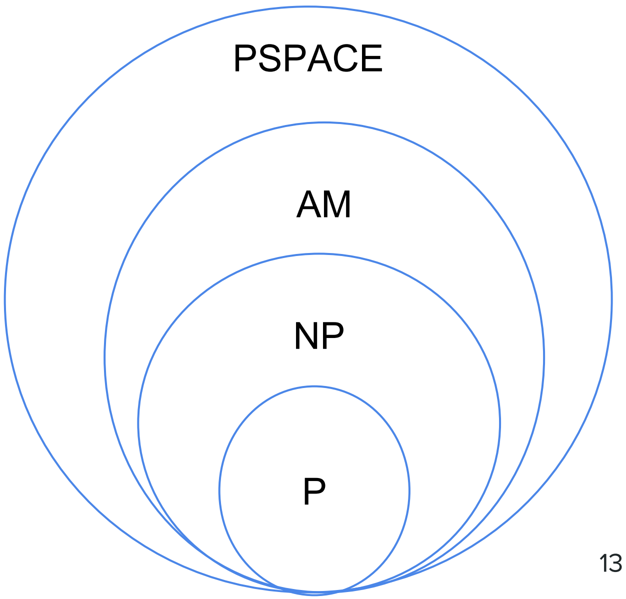
The goal is to construct a "constant-round protocol" for SAT by using the reduction.

Arthur-Merlin Protocol

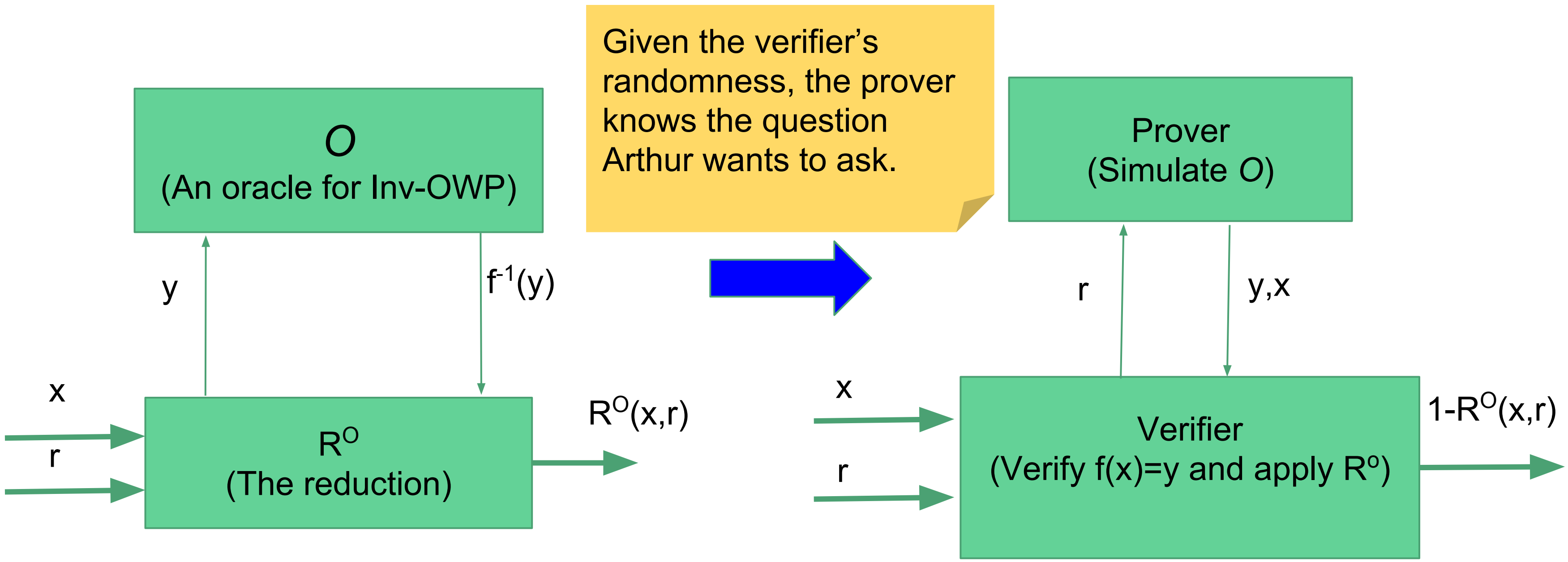


We say $L \in AM$ if

- (completeness) if $x \in L$, there is a prover (Merlin) can convince Arthur (the verifier) that $x \in L$.
- (soundness) if $x \notin L$, no prover (Merlin) can convince Arthur that $x \in L$.

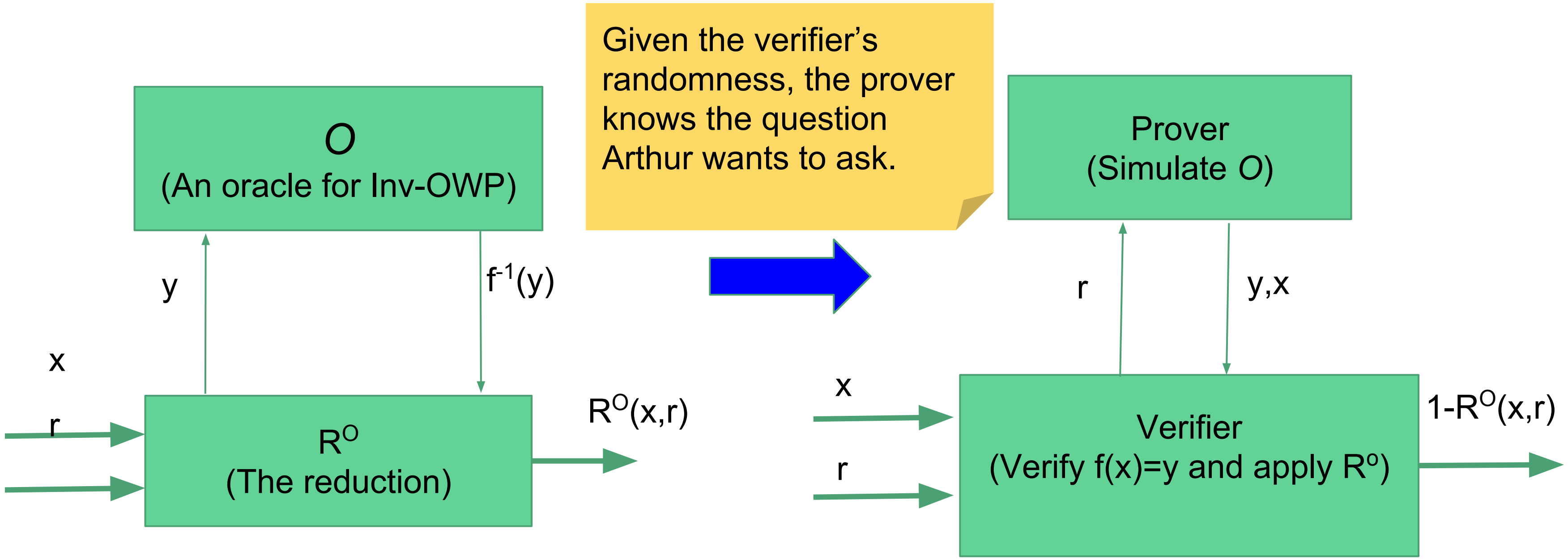


$SAT \leq_c$ Inv-OWP $\Rightarrow \overline{SAT} \in AM$



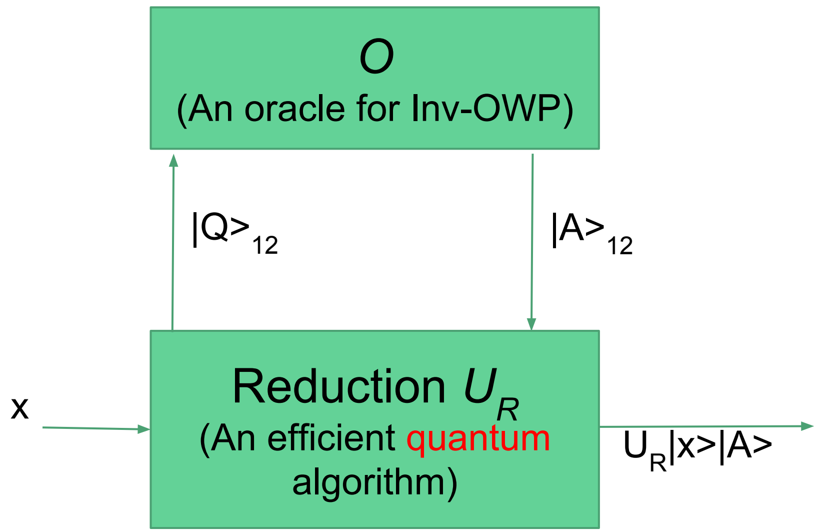
1. The verifier sends his random string to the prover.
   - The prover knows y after having the random string.
2. The prover sends y and x (where f(x)=y) to the verifier.
   - A malicious prover may send $(y', x') \neq (y, x)$.
3. The verifier verifies whether y is the question and $f(x) = y$. If not, reject.
4. The verifier runs the reduction R^o if he doesn't reject in step 3.

Can we use this protocol for quantum reductions?



1. The verifier sends his random string to the prover.
   - The prover knows y after having the random string.
2. The prover sends y and x (where f(x)=y) to the verifier.
   - A malicious prover may send (y', x') ≠ (y, x).
3. The verifier verifies whether y is the question and f(x) = y. If not, reject.
4. The verifier runs the reduction R^o if he doesn't reject in step 3.

No, quantum reductions are more tricky



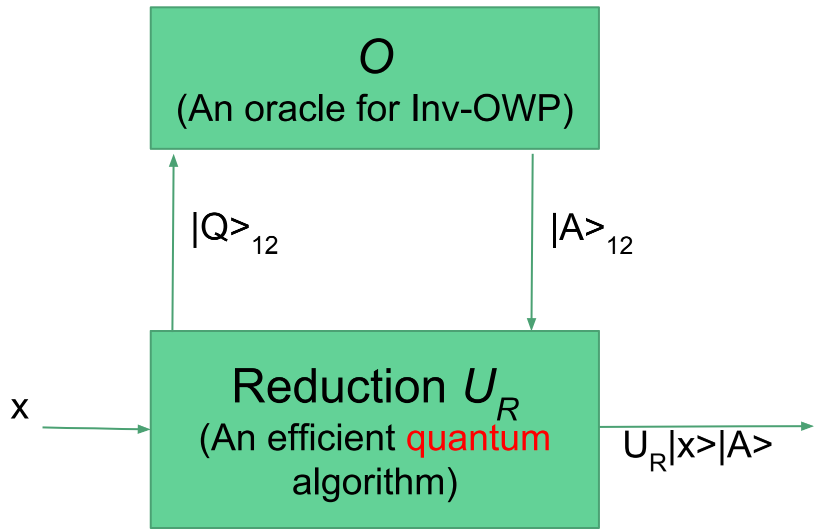
Each question can be in superposition

- $|Q>_{123}=\Sigma_q c_q|q>_1|0>_2|w_q>_3$
- $|c_q|^2$ can be viewed as the weight of question q.

The answer is also in superposition

- $|A>_{123}=\Sigma_q c_q|q>_1|f^{-1}(q)>_2|w_q>_3$

Why does the classical protocol fail?



Each question can be in superposition

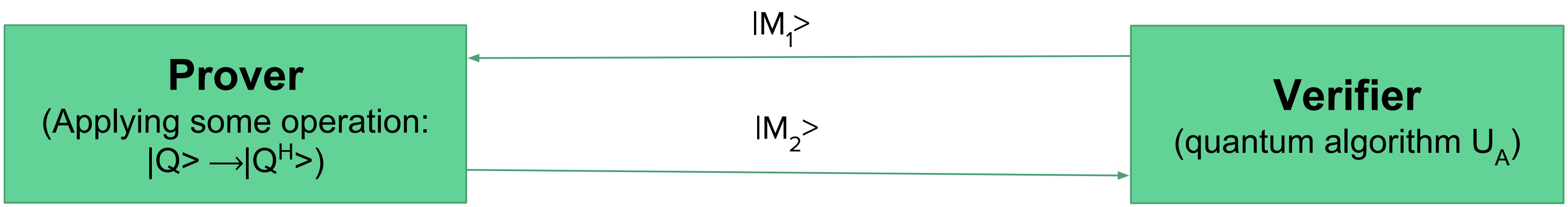
- $|Q>_{123}=\Sigma_q c_q|q>_1|0>_2|w_q>_3$
- $|c_q|^2$ can be viewed as the weight of question q.

The answer is also in superposition

- $|A>_{123}=\Sigma_q c_q|q>_1|f^{-1}(q)>_2|w_q>_3$

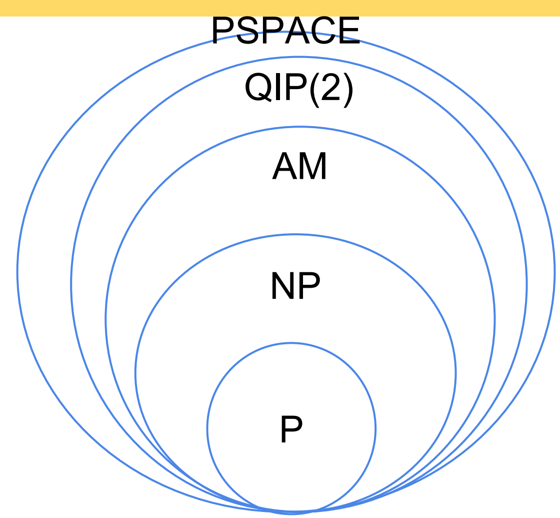
- SImulating the reduction SAT $\leq_q$ Inv-OWP only gives "quantum interactive proof" protocol.
- The prover can cheat by giving correct $(q,f^{-1}(q))$, but changing the weight c_q.

Goal: SAT $\leq_q$ Inv-OWP $\Rightarrow \overline{SAT} \in$ QIP(2)

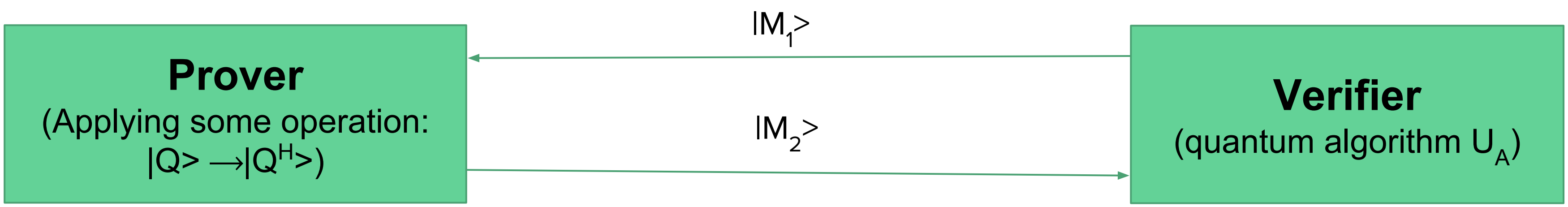


We say $L \in QIP(2)$ if

- (completeness) if $x \in L$, the prover can convince the verifier that $x \in L$.
- (soundness) if $x \notin L$, no prover can convince the verifier that $x \in L$.

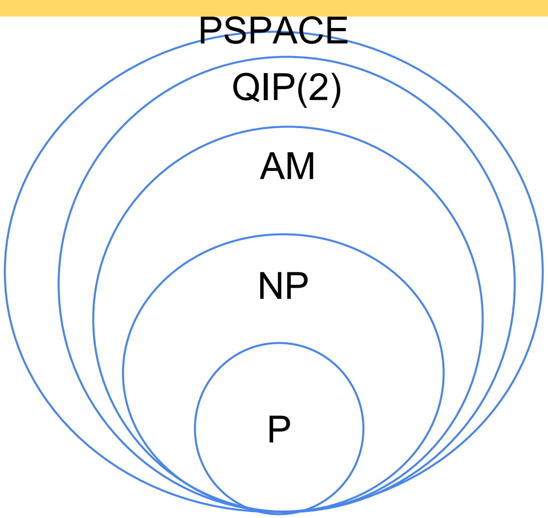


Goal: SAT $\le_q$ Inv-OWP $\Rightarrow$ $\overline{\text{SAT}} \in$ QIP(2) under uniform quantum reductions



We say $L \in QIP(2)$ if

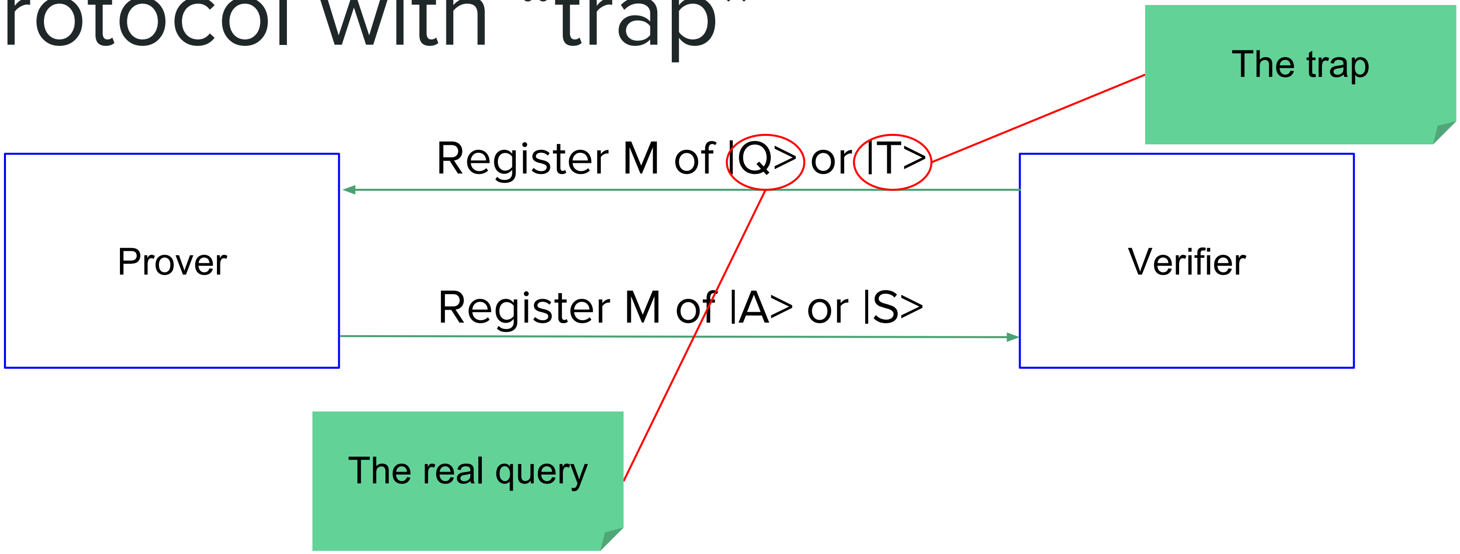
- (completeness) if $x \in L$, the prover can convince the verifier that $x \in L$.
- (soundness) if $x \notin L$, no prover can convince the verifier that $x \in L$.



Uniform quantum reductions:

- Each query is a uniform superposition
  - $|Q\rangle = \Sigma_q |q\rangle|0\rangle|w_q\rangle$
- The answer is also in uniform superposition
  - $|A\rangle = \Sigma |q\rangle|f^{-1}(q)\rangle|w_q\rangle$

A protocol with “trap”



The main idea: If the prover cheats, he has ½ probability to cheat on the trap state. The verifier can catch him by verifying the trap state!

- The prover cannot distinguish the trap and the real query.
- |S> can be efficiently verified by the verifier.

A protocol with “trap”

The trap

Register M of |Q> or |T>

Prover

Verifier

Register M of |A> or |S>

The real query

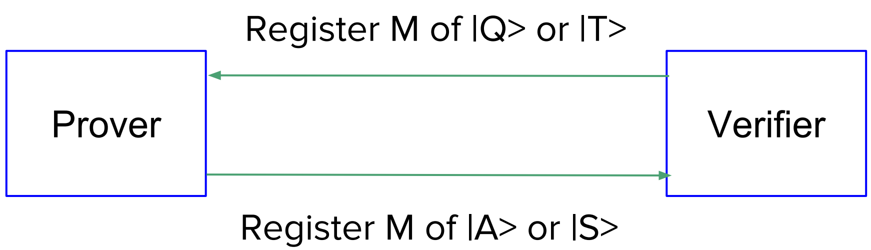
1. Send the register M of |Q> or |T> uniformly at random.
- $|Q\rangle=\Sigma_q(|q\rangle|0\rangle)_M(|w_q\rangle|q\rangle)_V$
- $|T\rangle=\Sigma_q(|q\rangle|0\rangle)_M(|0\rangle|q\rangle)_V$

2. An honest prover will send |A> or |S>.
- $|A\rangle=\Sigma_q|q\rangle|f^{-1}(q)\rangle|w_q\rangle|q\rangle$
- $|S\rangle=\Sigma_q|q\rangle|f^{-1}(q)\rangle|0\rangle|q\rangle$

3. The verifier does the following.
- In case |Q>:
  - Run the reduction and accept if the reduction accepts.
- In case |T>:
  - Run the unitary U: |S> ⇒ |0> and measure the output in the standard basis. If the outcome is |0>, accepts.

- |A> ⇒ |0> may not be efficient.
- U: |S> ⇒ |0> is efficient.

Analysis of the trap protocol



1. Send the register M of $|Q\rangle$ or $|T\rangle$ uniformly at random.

- $|Q\rangle=\Sigma_q(|q\rangle|0\rangle)_M(|w_q\rangle|q\rangle)_V$
- $|T\rangle=\Sigma_q(|q\rangle|0\rangle)_M(|0\rangle|q\rangle)_V$

3. The verifier does the following.

- In case $|Q\rangle$:
  - Run the reduction and accept if the reduction accepts.
- In case $|T\rangle$:
  - Run the unitary $U: |S\rangle \Rightarrow |0\rangle$ and measure the output in the standard basis. If the outcome is $|0\rangle$, accepts.

- The prover does not know which state he gets.
- No matter which operator the prover applies, it will
  - Change $|S\rangle$ a lot
    - Suppose $|S'\rangle$ is far from $|S\rangle$. By applying $U: |S\rangle \Rightarrow |0\ldots0\rangle$, $|S'\rangle$ is far from $|0\ldots0\rangle$.
  - Or changes $|A\rangle$ little.
    - Suppose $|A'\rangle \fallingdotseq |A\rangle$. By applying the reduction, $|A'\rangle$ will be rejected with high probability.

In these two cases, the verifier rejects with high probability.

Theorem: SAT$\leq_{uq}$ Inv-OWP $\Rightarrow$ coNP$\subseteq$QIP(2).

The result coNP$\subseteq$QIP(2) is not as strong as PH collapses, However, it is a nontrivial consequence of the existence of quantum reductions.

The “trap” protocol can be easily extended to quantum reductions with multiple non-adaptive queries.

We can deal with other non-uniform distributions which are not far from the uniform distribution by quantum resampling.

Open questions

- Can we deal with other distributions or adaptive queries?
- We shall revisit other no-go theorems for crypto primitives.
  - For cryptographic primitives which security are not based on NP-complete problems under classical reductions, can NP-complete problems reduce to them if quantum reductions are allowed?
  - E.g., Private information retrieval (PIR), FHE, Inv-OWF, …
- Can we give more evidences that coNP is not in QIP(2)?
- Can we find other consequence which is stronger than coNP $\subseteq$ QIP(2)?
  - E.g., coNP$\subseteq$QAM or QMA.
- Can we find a example where we can prove quantum reductions are more powerful than classical reductions?
- Generally, people think quantum algorithms make crypto systems less computationally secure. But, maybe it can make crypto systems securer by reducing hard problems to these systems.