Is it Easier to Prove Theorems that are Guaranteed to be True?

Rafael Pass & Muthuramakrishnan Venkitasubramaniam

Presented by Yizhi Huang & Jiaqian Li

2024-02-15

TL;DR

NO.

TL;DR (actual)

TFNP is hard on average in Pessiland.*

TL;DR (actual)

TFNP is hard on average in Pessiland.*
That is, if **NP** is hard on average and OWF don't exist, then **TFNP** is hard on average.

TL;DR (actual)

TFNP is hard on average in Pessiland.*
That is, if **NP** is hard on average and OWF don't exist, then **TFNP** is hard on average.

- In comparison, recall [Hubáček-Naor-Yogev'16] showed that if **NP** is hard on average, then **TFNP**/**poly** is hard on average.

TL;DR (actual)

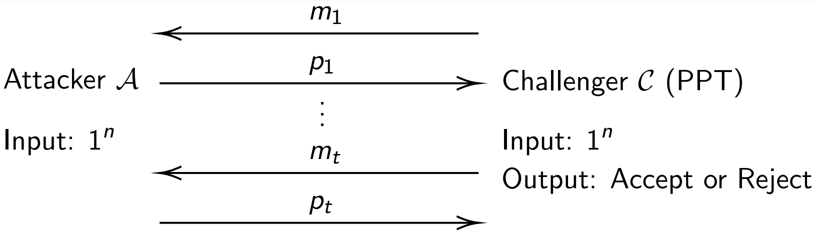
TFNP is hard on average in Pessiland.*

That is, if **NP** is hard on average and OWF don't exist, then **TFNP** is hard on average.

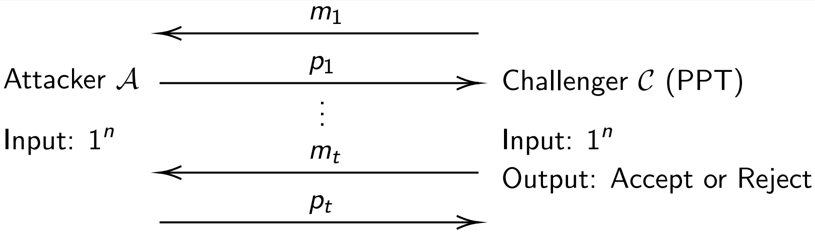
- In comparison, recall [Hubáček-Naor-Yogev'16] showed that if **NP** is hard on average, then **TFNP**/**poly** is hard on average.

Equivalently, if **NP** is hard on average, then either OWF exist, or **TFNP** is hard on average.

Interactive puzzles

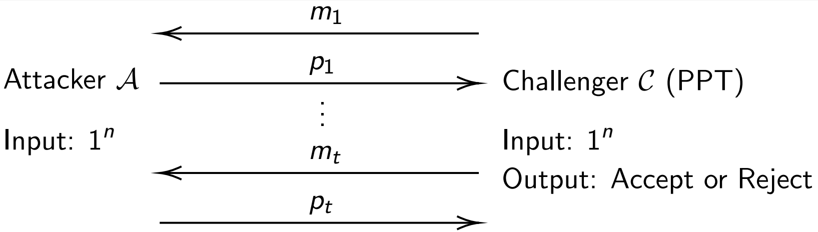


Interactive puzzles



- **Completeness.** There exists an (inefficient) attacker $\mathcal{A}(1^n)$ that succeeds in making $\mathcal{C}(1^n)$ accept unless with negligible probability.
- **Computational Soundness.** There does not exists PPT attacker $\mathcal{A}^*(1^n)$ that succeeds in making $\mathcal{C}(1^n)$ accept with inverse polynomial probability.
- **Public Verifiability.** Whether $\mathcal{C}(1^n)$ accepts is a deterministic poly-time function over the transcript $(m_1, p_1, \ldots, m_k, p_k)$.

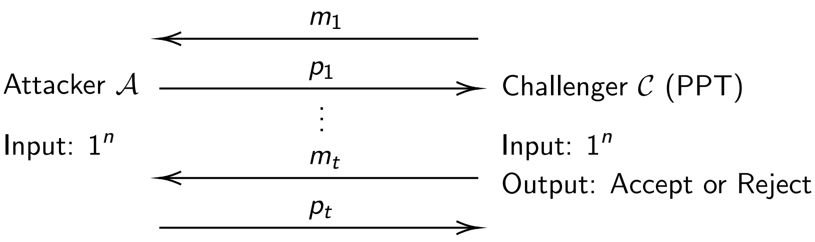
Interactive puzzles



- **Completeness.** There exists an (inefficient) attacker $\mathcal{A}(1^n)$ that succeeds in making $\mathcal{C}(1^n)$ accept unless with negligible probability.
- **Computational Soundness.** There does not exists PPT attacker $\mathcal{A}^*(1^n)$ that succeeds in making $\mathcal{C}(1^n)$ accept with inverse polynomial probability.
- **Public Verifiability.** Whether $\mathcal{C}(1^n)$ accepts is a deterministic poly-time function over the transcript $(m_1, p_1, \ldots, m_k, p_k)$.

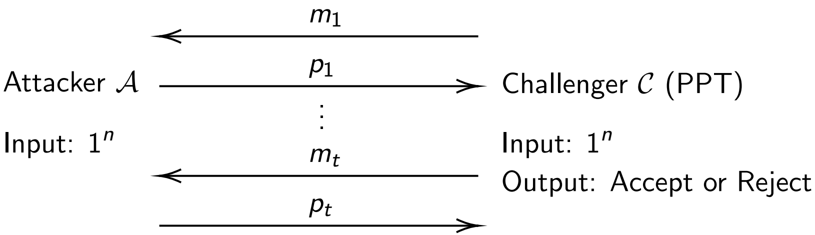
Remark. Negligible can be changed to $1/3$.

Interactive puzzles (optional properties)



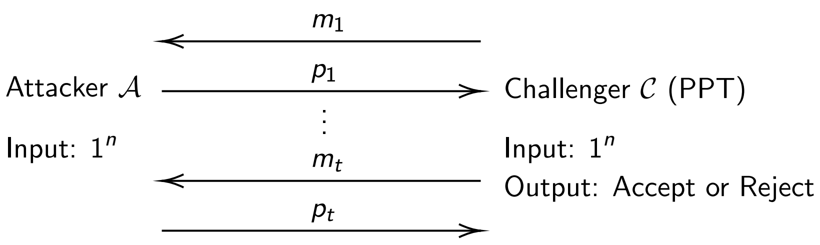
- k-**round** if the attacker and the challenger send k messages in total (for example, the above diagram is $2t$-round).

Interactive puzzles (optional properties)



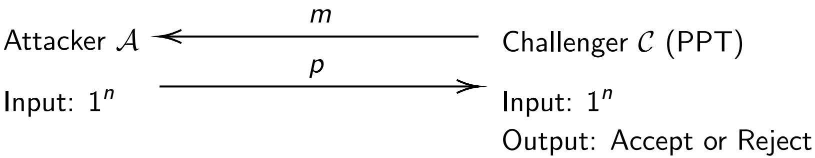
- *k*-**round** if the attacker and the challenger send k messages in total (for example, the above diagram is $2t$-round).
- **Public-coin** if the challenger only sends her randomness in each round. (The attacker can perform all computation instead.)

Interactive puzzles (optional properties)

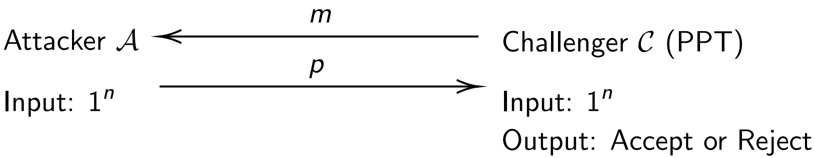


- k-**round** if the attacker and the challenger send k messages in total (for example, the above diagram is $2t$-round).
- **Public-coin** if the challenger only sends her randomness in each round. (The attacker can perform all computation instead.)
- **Perfect completeness** if there exists an attacker $\mathcal{A}$ that always succeeds in making $\mathcal{C}(1^n)$ output 1.

2-round puzzles

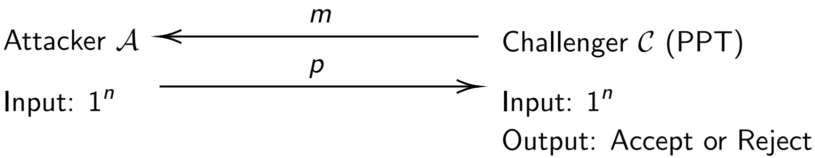


2-round puzzles



(m, p) is an **NP** relation (because of public-verifiability).

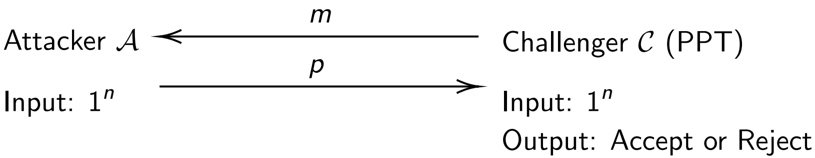
2-round puzzles



(m, p) is an **NP** relation (because of public-verifiability).

- The existence of a 2-round puzzle is syntactically equivalent to the existence of a hard-on-average search problem in **NP**.

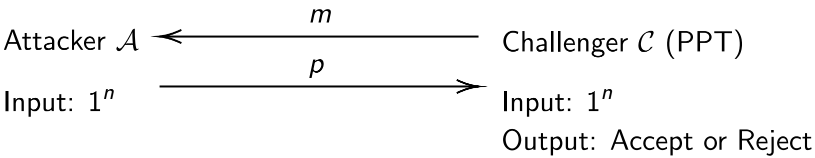
2-round puzzles



(m, p) is an **NP** relation (because of public-verifiability).

- The existence of a 2-round puzzle is syntactically equivalent to the existence of a hard-on-average search problem in **NP**.
- Public-coin iff the hard distribution is the uniform distribution.

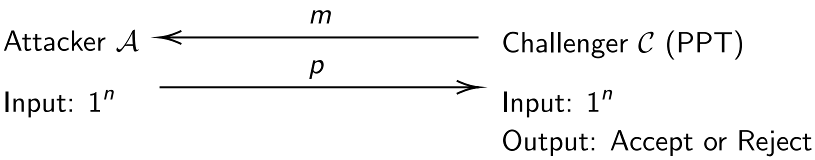
2-round puzzles



(m, p) is an **NP** relation (because of public-verifiability).

- The existence of a 2-round puzzle is syntactically equivalent to the existence of a hard-on-average search problem in **NP**.
- Public-coin iff the hard distribution is the uniform distribution.
- Perfect-completeness iff the problem is *promise-true*.
(Promise-true here means we restrict the problem the instances that have a solution, but does not mean the search problem is total.)

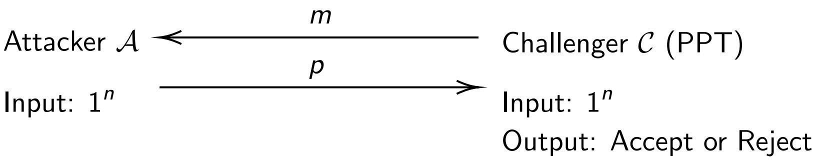
2-round puzzles



(m, p) is an **NP** relation (because of public-verifiability).

- The existence of a 2-round puzzle is syntactically equivalent to the existence of a hard-on-average search problem in **NP**.
- Public-coin iff the hard distribution is the uniform distribution.
- Perfect-completeness iff the problem is *promise-true*.
  (Promise-true here means we restrict the problem the instances that have a solution, but does not mean the search problem is total. Examples include **TFNP** and inverting OWF.)

2-round puzzles



(m, p) is an **NP** relation (because of public-verifiability).

- The existence of a 2-round puzzle is syntactically equivalent to the existence of a hard-on-average search problem in **NP**.
- Public-coin iff the hard distribution is the uniform distribution.
- Perfect-completeness iff the problem is *promise-true*. (Promise-true here means we restrict the problem the instances that have a solution, but does not mean the search problem is total. Examples include **TFNP** and inverting OWF.)
- If the puzzle is both public-coin and perfectly complete, then the hard-on-average problem is in **TFNP**.

Comparison to interactive proofs

- In interactive proofs, the verifier and prover get an instance x of a language L, but in puzzles, the attacker and challenger do not.

Comparison to interactive proofs

- In interactive proofs, the verifier and prover get an instance x of a language L, but in puzzles, the attacker and challenger do not.
- In interactive proofs, the prover for soundness can be computationally unbounded, but in puzzles, the attacker for soundness is computationally bounded.

Comparison to interactive proofs

- In interactive proofs, the verifier and prover get an instance x of a language L, but in puzzles, the attacker and challenger do not.
- In interactive proofs, the prover for soundness can be computationally unbounded, but in puzzles, the attacker for soundness is computationally bounded.
- In interactive proofs, the difference between completeness and soundness arises from whether $x \in L$, whereas in puzzles, it arises from the difference in the computation power of attackers.

Proof overview

Main result. If **NP** is hard on average and OWF don't exist, then **TFNP** is hard on average.

Proof overview

Main result. If **NP** is hard on average and OWF don't exist, then **TFNP** is hard on average.

NP is hard on average

$\Downarrow$

There exists a 2-round public-coin puzzle

Proof overview

Main result. If **NP** is hard on average and OWF don't exist, then **TFNP** is hard on average.

NP is hard on average

$\Downarrow$

There exists a 2-round public-coin puzzle

$\Downarrow$

There exists a 3-round public-coin puzzle with perfect completeness

Proof overview

Main result. If **NP** is hard on average and OWF don't exist, then **TFNP** is hard on average.

NP is hard on average

$\Downarrow$

There exists a 2-round public-coin puzzle

$\Downarrow$

There exists a 3-round public-coin puzzle with perfect completeness

$\Downarrow$ (Assume OWF don't exist)

There exists a 2-round public-coin puzzle with perfect completeness

Proof overview

Main result. If **NP** is hard on average and OWF don't exist, then **TFNP** is hard on average.

NP is hard on average

$\Downarrow$

There exists a 2-round public-coin puzzle

$\Downarrow$

There exists a 3-round public-coin puzzle with perfect completeness

$\Downarrow$ (Assume OWF don't exist)

There exists a 2-round public-coin puzzle with perfect completeness

$\Downarrow$

TFNP is hard on average

Proof overview

Main result. If **NP** is hard on average and OWF don't exist, then **TFNP** is hard on average.

NP is hard on average

$\Downarrow$

There exists a 2-round public-coin puzzle

$\Downarrow$

There exists a 3-round public-coin puzzle with perfect completeness

$\Downarrow$ (Assume OWF don't exist)

There exists a 2-round public-coin puzzle with perfect completeness

$\Downarrow$

TFNP is hard on average

Step 1/4: from hard-on-average problems to puzzles

We want to prove:

NP is hard on average $\Longrightarrow$ There exists a 2-round public-coin puzzle

Step 1/4: from hard-on-average problems to puzzles

We want to prove:
NP is hard on average $\Longrightarrow$ There exists a 2-round public-coin puzzle

Lemma. If an **NP** problem L is hard on an efficiently-samplable distribution $\mathcal{D}$, then there exists an **NP** problem L' that is hard on the uniform distribution.

Proof overview

Main result. If **NP** is hard on average and OWF don't exist, then **TFNP** is hard on average.

NP is hard on average

✓$\Downarrow$

There exists a 2-round public-coin puzzle

$\Downarrow$

There exists a 3-round public-coin puzzle with perfect completeness

$\Downarrow$ (Assume OWF don't exist)

There exists a 2-round public-coin puzzle with perfect completeness

$\Downarrow$

TFNP is hard on average

Step 2/4: perfect completeness at the expense of a round

We want to prove:

a 2-round public-coin puzzle $\Longrightarrow$

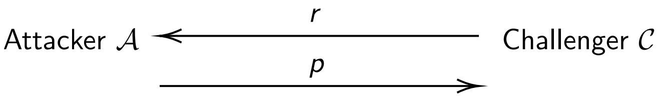
a 3-round public-coin puzzle with perfect completeness

Step 2/4: perfect completeness at the expense of a round

We want to prove:

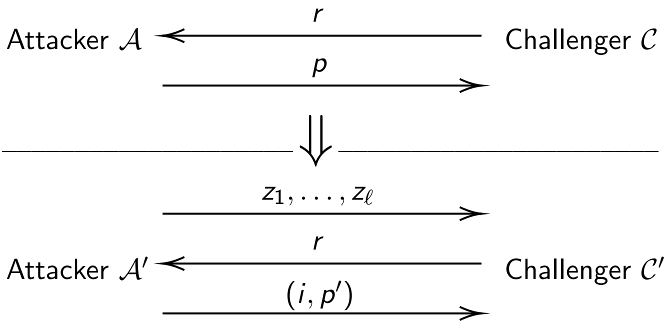
a 2-round public-coin puzzle $\Longrightarrow$

a 3-round public-coin puzzle with perfect completeness



Step 2/4: perfect completeness at the expense of a round

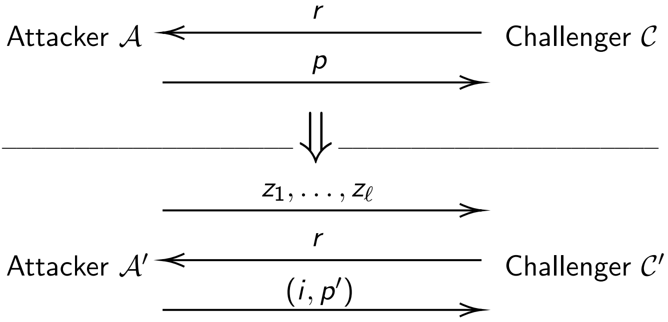
We want to prove:
a 2-round public-coin puzzle $\Longrightarrow$
a 3-round public-coin puzzle with perfect completeness



$\mathcal{C}'$ accepts iff $\mathcal{C}(z_i \oplus r, p') = 1$.

Step 2/4: perfect completeness at the expense of a round

We want to prove:
a 2-round public-coin puzzle $\Longrightarrow$
a 3-round public-coin puzzle with perfect completeness



$\mathcal{C}'$ accepts iff $\mathcal{C}(z_i \oplus r, p') = 1$.

It can be proven that there exists a way to select $z_1, \ldots, z_\ell$ such that the completeness is perfect and the soundness still holds.

Proof overview

Main result. If **NP** is hard on average and OWF don't exist, then **TFNP** is hard on average.

NP is hard on average

✓⇓

There exists a 2-round public-coin puzzle

✓⇓

There exists a 3-round public-coin puzzle with perfect completeness

⇓ (Assume OWF don't exist)

There exists a 2-round public-coin puzzle with perfect completeness

⇓

TFNP is hard on average

Step 3/4: round reduction

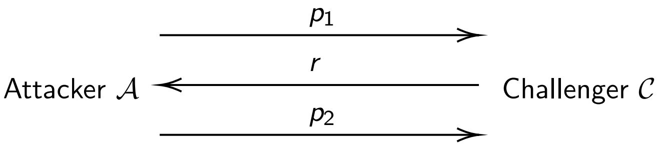
We want to prove:
Assuming OWF don't exist,
a 3-round public-coin puzzle with perfect completeness $\Longrightarrow$
a 2-round public-coin puzzle with perfect completeness

Step 3/4: round reduction

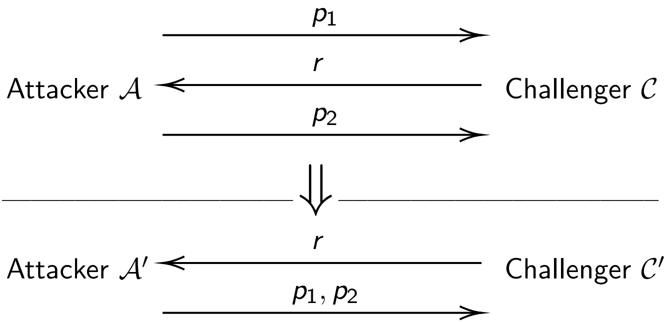
We want to prove:
Assuming OWF don't exist,
a 3-round public-coin puzzle with perfect completeness $\Longrightarrow$
a 2-round public-coin puzzle with perfect completeness

The proof actually works for k-round to $(k-1)$-round for any polynomial $k(n)$. For simplicity, we only consider $k = 3$.

First attempt

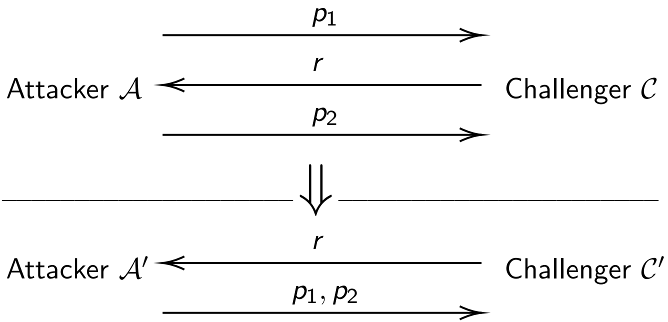


First attempt



$\mathcal{C}'$ accepts iff $\mathcal{C}(p_1, r, p_2) = 1$.

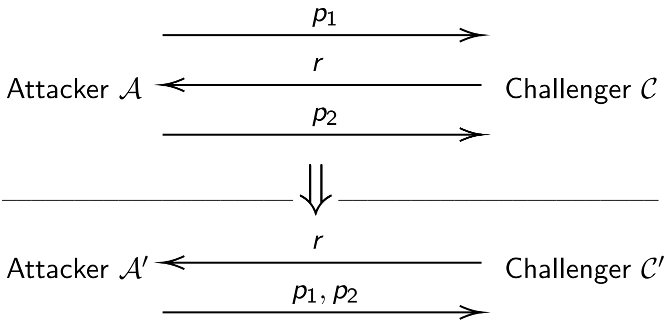
First attempt



$\mathcal{C}'$ accepts iff $\mathcal{C}(p_1, r, p_2) = 1$.

Perfect completeness. Trivial.

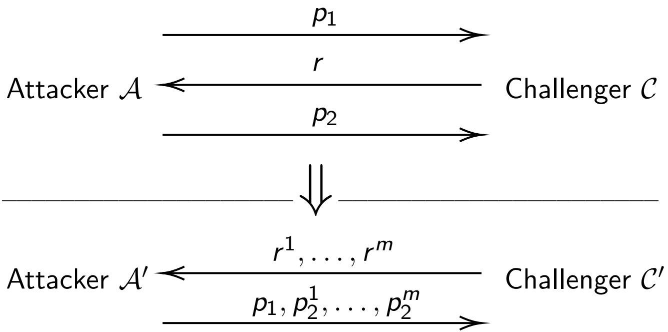
First attempt



$\mathcal{C}'$ accepts iff $\mathcal{C}(p_1, r, p_2) = 1$.

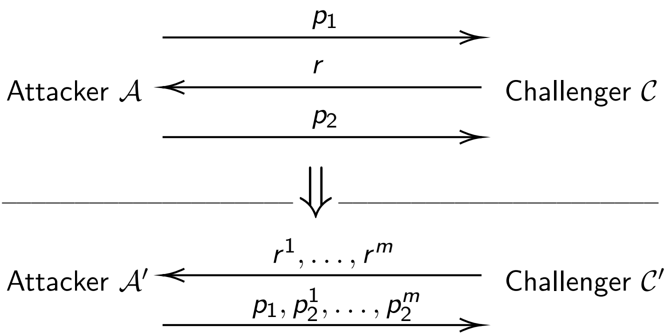
Perfect completeness. Trivial.
Soundness. False.

[Babai-Moran'88] round reduction



$\mathcal{C}'$ accepts iff $\mathcal{C}(p_1, r^i, p_2^i) = 1$ for all i.

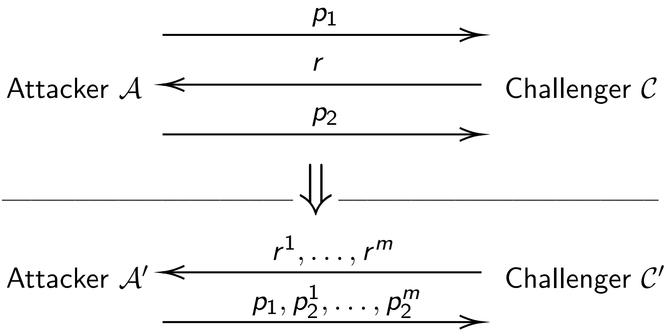
[Babai-Moran'88] round reduction



$\mathcal{C}'$ accepts iff $\mathcal{C}(p_1, r^i, p_2^i) = 1$ for all i.

Perfect completeness. Trivial.

[Babai-Moran'88] round reduction

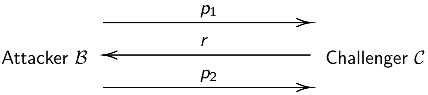


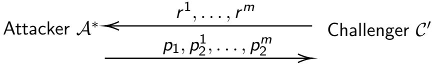
$\mathcal{C}'$ accepts iff $\mathcal{C}(p_1, r^i, p_2^i) = 1$ for all i.

Perfect completeness. Trivial.

Soudness. [BM88] showed that the transformation preserves soundness in their context of computationally-unbounded $\mathcal{A}, \mathcal{A}'$, but in our setting, soundness is for PPT $\mathcal{A}, \mathcal{A}'$.

Soundness of the round reduction (informal)

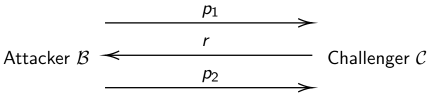


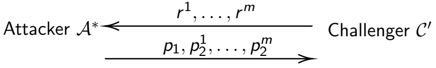


$\mathcal{C}'$ accepts iff $\mathcal{C}(p_1, r^i, p_2^i) = 1$ for all i.

Suppose a PPT $\mathcal{A}^*$ breaks the soundness of the 2-round $\mathcal{C}'$, we construct a PPT $\mathcal{B}$ that breaks the soundness of the 3-round $\mathcal{C}$.

Soundness of the round reduction (informal)



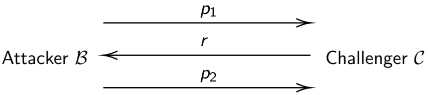


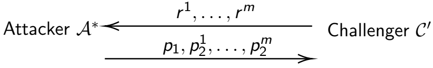
$\mathcal{C}'$ accepts iff $\mathcal{C}(p_1, r^i, p_2^i) = 1$ for all i.

Suppose a PPT $\mathcal{A}^*$ breaks the soundness of the 2-round $\mathcal{C}'$, we construct a PPT $\mathcal{B}$ that breaks the soundness of the 3-round $\mathcal{C}$.

$\mathcal{B}$ has randomness $s = (s^1, \ldots, s^m, z)$. ($s^i$ are supposed to be the messages $\mathcal{A}^*$ receive, and z the randomness of $\mathcal{A}^*$.)
On the first round, $\mathcal{B}$ simulates $(p_1, p_2^1, \ldots, p_2^m) := \mathcal{A}^*(s)$ and outputs p_1.

Soundness of the round reduction (informal)





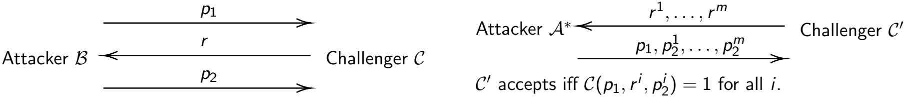
$\mathcal{C}'$ accepts iff $\mathcal{C}(p_1, r^i, p_2^i) = 1$ for all i.

Suppose a PPT $\mathcal{A}^*$ breaks the soundness of the 2-round $\mathcal{C}'$, we construct a PPT $\mathcal{B}$ that breaks the soundness of the 3-round $\mathcal{C}$.

$\mathcal{B}$ has randomness $s = (s^1, \ldots, s^m, z)$. ($s^i$ are supposed to be the messages $\mathcal{A}^*$ receive, and z the randomness of $\mathcal{A}^*$.)
On the first round, $\mathcal{B}$ simulates $(p_1, p_2^1, \ldots, p_2^m) := \mathcal{A}^*(s)$ and outputs p_1.

On the third round, suppose $\mathcal{B}$ receives r from $\mathcal{C}$.

Soundness of the round reduction (informal)

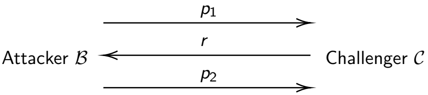


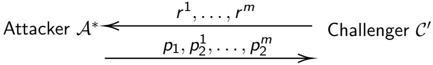
Suppose a PPT $\mathcal{A}^*$ breaks the soundness of the 2-round $\mathcal{C}'$, we construct a PPT $\mathcal{B}$ that breaks the soundness of the 3-round $\mathcal{C}$.

$\mathcal{B}$ has randomness $s = (s^1, \ldots, s^m, z)$. ($s^i$ are supposed to be the messages $\mathcal{A}^*$ receive, and z the randomness of $\mathcal{A}^*$.)
On the first round, $\mathcal{B}$ simulates $(p_1, p_2^1, \ldots, p_2^m) := \mathcal{A}^*(s)$ and outputs p_1.

On the third round, suppose $\mathcal{B}$ receives r from $\mathcal{C}$. If $r = s^i$ for some i, $\mathcal{B}$ can output p_2^i.

Soundness of the round reduction (informal)





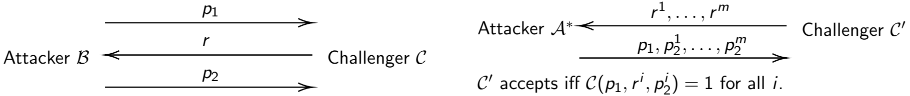
$\mathcal{C}'$ accepts iff $\mathcal{C}(p_1, r^i, p_2^i) = 1$ for all i.

Suppose a PPT $\mathcal{A}^*$ breaks the soundness of the 2-round $\mathcal{C}'$, we construct a PPT $\mathcal{B}$ that breaks the soundness of the 3-round $\mathcal{C}$.

$\mathcal{B}$ has randomness $s = (s^1, \ldots, s^m, z)$. ($s^i$ are supposed to be the messages $\mathcal{A}^*$ receive, and z the randomness of $\mathcal{A}^*$.)
On the first round, $\mathcal{B}$ simulates $(p_1, p_2^1, \ldots, p_2^m) := \mathcal{A}^*(s)$ and outputs p_1.

On the third round, suppose $\mathcal{B}$ receives r from $\mathcal{C}$. If $r = s^i$ for some i, $\mathcal{B}$ can output p_2^i. But what if $r \notin \{s^1, \ldots, s^m\}$?

Soundness of the round reduction (informal)



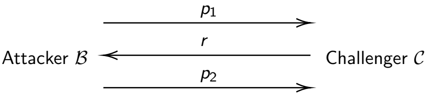
$\mathcal{C}'$ accepts iff $\mathcal{C}(p_1, r^i, p_2^i) = 1$ for all i.

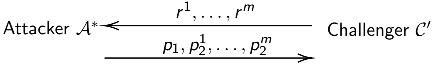
We construct a PPT $\mathcal{B}$ from the PPT $\mathcal{A}^*$.

$\mathcal{B}$ has randomness $s = (s^1, \ldots, s^m, z)$.
On the first round, $\mathcal{B}$ simulates $(p_1, p_2^1, \ldots, p_2^m) := \mathcal{A}^*(s)$ and outputs p_1.
On the third round, suppose $\mathcal{B}$ receives r from $\mathcal{C}$. If $r = s^i$ for some i, $\mathcal{B}$ can output p_2^i. But what if $r \notin \{s^1, \ldots, s^m\}$?

We want to find another transcript of $\mathcal{A}^*$ and $\mathcal{C}'$ in which p_1, r appear.

Soundness of the round reduction (informal)





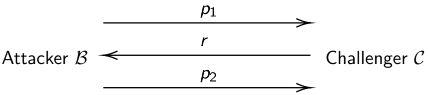
$\mathcal{C}'$ accepts iff $\mathcal{C}(p_1, r^i, p_2^i) = 1$ for all i.

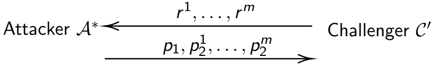
We construct a PPT $\mathcal{B}$ from the PPT $\mathcal{A}^*$.

$\mathcal{B}$ has randomness $s = (s^1, \ldots, s^m, z)$.
On the first round, $\mathcal{B}$ simulates $(p_1, p_2^1, \ldots, p_2^m) := \mathcal{A}^*(s)$ and outputs p_1.
On the third round, suppose $\mathcal{B}$ receives r from $\mathcal{C}$. If $r = s^i$ for some i, $\mathcal{B}$ can output p_2^i. But what if $r \notin \{s^1, \ldots, s^m\}$?
We want to find another transcript of $\mathcal{A}^*$ and $\mathcal{C}'$ in which p_1, r appear.

A transcript of $\mathcal{A}^*$ and $\mathcal{C}'$ is a function of the randomness z of $\mathcal{A}^*$ and $(r^1, \ldots, r^m)$ of $\mathcal{C}'$. Thus, (p_1, r^i) is a function (denoted M) of them and i.

Soundness of the round reduction (informal)





$\mathcal{C}'$ accepts iff $\mathcal{C}(p_1, r^i, p_2^i) = 1$ for all i.

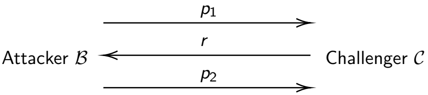
We construct a PPT $\mathcal{B}$ from the PPT $\mathcal{A}^*$.

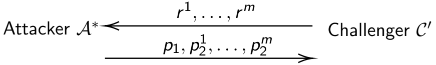
$\mathcal{B}$ has randomness $s = (s^1, \ldots, s^m, z)$.
On the first round, $\mathcal{B}$ simulates $(p_1, p_2^1, \ldots, p_2^m) := \mathcal{A}^*(s)$ and outputs p_1.
On the third round, suppose $\mathcal{B}$ receives r from $\mathcal{C}$. If $r = s^i$ for some i, $\mathcal{B}$ can output p_2^i. But what if $r \notin \{s^1, \ldots, s^m\}$?
We want to find another transcript of $\mathcal{A}^*$ and $\mathcal{C}'$ in which p_1, r appear.

A transcript of $\mathcal{A}^*$ and $\mathcal{C}'$ is a function of the randomness z of $\mathcal{A}^*$ and $(r^1, \ldots, r^m)$ of $\mathcal{C}'$. Thus, (p_1, r^i) is a function (denoted M) of them and i.

$\mathcal{B}$ gets $(j, t^1, \ldots, t^m, z') := Inv(p_1, r)$ where Inv inverts M. (If Inv succeeds, then $t^j = r$.)

Soundness of the round reduction (informal)





$\mathcal{C}'$ accepts iff $\mathcal{C}(p_1, r^i, p_2^i) = 1$ for all i.

We construct a PPT $\mathcal{B}$ from the PPT $\mathcal{A}^*$.

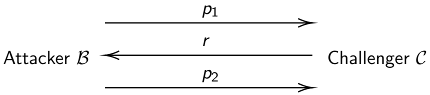
$\mathcal{B}$ has randomness $s = (s^1, \ldots, s^m, z)$.
On the first round, $\mathcal{B}$ simulates $(p_1, p_2^1, \ldots, p_2^m) := \mathcal{A}^*(s)$ and outputs p_1.
On the third round, suppose $\mathcal{B}$ receives r from $\mathcal{C}$. If $r = s^i$ for some i, $\mathcal{B}$ can output p_2^i. But what if $r \notin \{s^1, \ldots, s^m\}$?
We want to find another transcript of $\mathcal{A}^*$ and $\mathcal{C}'$ in which p_1, r appear.

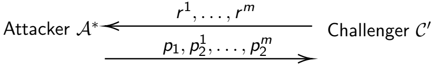
A transcript of $\mathcal{A}^*$ and $\mathcal{C}'$ is a function of the randomness z of $\mathcal{A}^*$ and $(r^1, \ldots, r^m)$ of $\mathcal{C}'$. Thus, (p_1, r^i) is a function (denoted M) of them and i.

$\mathcal{B}$ gets $(j, t^1, \ldots, t^m, z') := Inv(p_1, r)$ where Inv inverts M. (If Inv succeeds, then $t^j = r$.)

Then, $\mathcal{B}$ lets $(q_1, q_2^1, \ldots, q_2^m) := \mathcal{A}^*(t^1, \ldots, t^m, z')$ and outputs q_2^j.

Soundness of the round reduction (informal)





$\mathcal{C}'$ accepts iff $\mathcal{C}(p_1, r^i, p_2^i) = 1$ for all i.

We construct a PPT $\mathcal{B}$ from the PPT $\mathcal{A}^*$.

$\mathcal{B}$ has randomness $s = (s^1, \ldots, s^m, z)$.
On the first round, $\mathcal{B}$ simulates $(p_1, p_2^1, \ldots, p_2^m) := \mathcal{A}^*(s)$ and outputs p_1.
On the third round, suppose $\mathcal{B}$ receives r from $\mathcal{C}$. If $r = s^i$ for some i, $\mathcal{B}$ can output p_2^i. But what if $r \notin \{s^1, \ldots, s^m\}$?
We want to find another transcript of $\mathcal{A}^*$ and $\mathcal{C}'$ in which p_1, r appear.

A transcript of $\mathcal{A}^*$ and $\mathcal{C}'$ is a function of the randomness z of $\mathcal{A}^*$ and $(r^1, \ldots, r^m)$ of $\mathcal{C}'$. Thus, (p_1, r^i) is a function (denoted M) of them and i.

$\mathcal{B}$ gets $(j, t^1, \ldots, t^m, z') := Inv(p_1, r)$ where Inv inverts M. (If Inv succeeds, then $t^j = r$.)
Then, $\mathcal{B}$ lets $(q_1, q_2^1, \ldots, q_2^m) := \mathcal{A}^*(t^1, \ldots, t^m, z')$ and outputs q_2^j.
If $\mathcal{A}^*$ and Inv both succeed, then $\mathcal{C}(p_1, r, q_2^j) = \mathcal{C}(p_1, t^j, q_2^j) = 1$.

Soundness of the round reduction (informal, cont'd)

In the last round, $\mathcal{B}$ uses the inverter *Inv* to produce a transcript of $\mathcal{A}^*$ and $\mathcal{C}'$ that is consistent with (p_1, r), and uses the output of $\mathcal{A}^*$ corresponding to r as the output of itself.

Soundness of the round reduction (informal, cont'd)

In the last round, $\mathcal{B}$ uses the inverter *Inv* to produce a transcript of $\mathcal{A}^*$ and $\mathcal{C}'$ that is consistent with (p_1, r), and uses the output of $\mathcal{A}^*$ corresponding to r as the output of itself.

If $\mathcal{A}^*$ and *Inv* both succeed, then $\mathcal{B}$ succeeds.
But they don't always succeed!

Soundness of the round reduction (informal, cont'd)

In the last round, $\mathcal{B}$ uses the inverter *Inv* to produce a transcript of $\mathcal{A}^*$ and $\mathcal{C}'$ that is consistent with (p_1, r), and uses the output of $\mathcal{A}^*$ corresponding to r as the output of itself.

If $\mathcal{A}^*$ and *Inv* both succeed, then $\mathcal{B}$ succeeds.
But they don't always succeed!

- The inverter should take inputs from a correct distribution.

Soundness of the round reduction (informal, cont'd)

In the last round, $\mathcal{B}$ uses the inverter *Inv* to produce a transcript of $\mathcal{A}^*$ and $\mathcal{C}'$ that is consistent with (p_1, r), and uses the output of $\mathcal{A}^*$ corresponding to r as the output of itself.

If $\mathcal{A}^*$ and *Inv* both succeed, then $\mathcal{B}$ succeeds.
But they don't always succeed!

- The inverter should take inputs from a correct distribution. Complicated, omitted.

Soundness of the round reduction (informal, cont'd)

In the last round, $\mathcal{B}$ uses the inverter *Inv* to produce a transcript of $\mathcal{A}^*$ and $\mathcal{C}'$ that is consistent with (p_1, r), and uses the output of $\mathcal{A}^*$ corresponding to r as the output of itself.

If $\mathcal{A}^*$ and *Inv* both succeed, then $\mathcal{B}$ succeeds.
But they don't always succeed!

- The inverter should take inputs from a correct distribution. Complicated, omitted.
- The inverter should produce a distribution that has low correlation with whether $\mathcal{A}^*$ succeeds.

Soundness of the round reduction (informal, cont'd)

In the last round, $\mathcal{B}$ uses the inverter *Inv* to produce a transcript of $\mathcal{A}^*$ and $\mathcal{C}'$ that is consistent with (p_1, r), and uses the output of $\mathcal{A}^*$ corresponding to r as the output of itself.

If $\mathcal{A}^*$ and *Inv* both succeed, then $\mathcal{B}$ succeeds.
But they don't always succeed!

- The inverter should take inputs from a correct distribution.
  Complicated, omitted.
- The inverter should produce a distribution that has low correlation with whether $\mathcal{A}^*$ succeeds.
  Use distributional OWF.

Soundness of the round reduction (informal, cont'd)

In the last round, $\mathcal{B}$ uses the inverter *Inv* to produce a transcript of $\mathcal{A}^*$ and $\mathcal{C}'$ that is consistent with (p_1, r), and uses the output of $\mathcal{A}^*$ corresponding to r as the output of itself.

If $\mathcal{A}^*$ and *Inv* both succeed, then $\mathcal{B}$ succeeds.
But they don't always succeed!

- The inverter should take inputs from a correct distribution.
  Complicated, omitted.
- The inverter should produce a distribution that has low correlation with whether $\mathcal{A}^*$ succeeds.
  Use distributional OWF.

Distributional OWF

f is a distributional OWF if is is hard to sample a uniformly random pre-image.
That is, for any PPT T,

$$\{(T(f(x)), f(x)) : x \leftarrow \{0,1\}^n\} \not\approx_s \{(x, f(x)) : x \leftarrow \{0,1\}^n\}.$$

Distributional OWF

f is a distributional OWF if is is hard to sample a uniformly random pre-image.
That is, for any PPT T,

$$\{(T(f(x)), f(x)) : x \leftarrow \{0,1\}^n\} \not\approx_s \{(x, f(x)) : x \leftarrow \{0,1\}^n\}.$$

Lemma. Existence of distributional OWF implies existence of OWF.

Soundness of the round reduction (informal, cont'd)

In the last round, $\mathcal{B}$ uses the inverter *Inv* to produce a transcript of $\mathcal{A}^*$ and $\mathcal{C}'$ that is consistent with (p_1, r), and uses the output of $\mathcal{A}^*$ corresponding to r as the output of itself.

If $\mathcal{A}^*$ and *Inv* both succeed, then $\mathcal{B}$ succeeds.
But they don't always succeed!

- The inverter should take inputs from a correct distribution.
  Complicated, omitted.
- The inverter should produce a distribution that has low correlation with whether $\mathcal{A}^*$ succeeds.
  Use distributional OWF.

Proof overview

Main result. If **NP** is hard on average and OWF don't exist, then **TFNP** is hard on average.

NP is hard on average

✓⇓

There exists a 2-round public-coin puzzle

✓⇓

There exists a 3-round public-coin puzzle with perfect completeness

✓⇓ (Assume OWF don't exist)

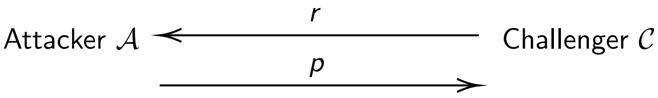
There exists a 2-round public-coin puzzle with perfect completeness

⇓

TFNP is hard on average

Step 4/4: **TFNP**-hardness-on-average from puzzles

There exists a 2-round public-coin puzzle with perfect completeness $\Longrightarrow$ **TFNP** is hard on average



Straight-forward from definition.

Proof overview

Main result. If **NP** is hard on average and OWF don't exist, then **TFNP** is hard on average.

NP is hard on average

✓⇓

There exists a 2-round public-coin puzzle

✓⇓

There exists a 3-round public-coin puzzle with perfect completeness

✓⇓ (Assume OWF don't exist)

There exists a 2-round public-coin puzzle with perfect completeness

✓⇓

TFNP is hard on average

Proof overview

Main result. If **NP** is hard on average and OWF don't exist, then **TFNP** is hard on average.

NP is hard on average

✓⇓

There exists a 2-round public-coin puzzle

✓⇓

There exists a 3-round public-coin puzzle with perfect completeness

✓⇓ (Assume OWF don't exist)

There exists a 2-round public-coin puzzle with perfect completeness

✓⇓

TFNP is hard on average

A caveat: infinitely-often

Main result: **TFNP** is hard on average in Pessiland.*

A caveat: infinitely-often

Main result: **TFNP** is hard on average in Pessiland.*

What we actually proved in the round-reduction step is, for every n, if there exists a 3-round puzzle (with some properties) with security parameter 1^n, then there exist either OWF with security parameter 1^n, or 2-round puzzles with security parameter 1^n.

A caveat: infinitely-often

Main result: **TFNP** is hard on average in Pessiland.*

What we actually proved in the round-reduction step is, for every n, if there exists a 3-round puzzle (with some properties) with security parameter 1^n, then there exist either OWF with security parameter 1^n, or 2-round puzzles with security parameter 1^n.
Therefore, even if 3-round puzzles exist for all sufficiently large n, we can only get the following:

- Either OWF exist for all sufficiently large n, or 2-round puzzles exist for infinitely many n.
- Either OWF exist for infinitely many n, or 2-round puzzles exist for all sufficiently large n.

Wait! But the title is. . . ?

Is it Easier to Prove Theorems that are Guaranteed to be True?

Rafael Pass & Muthuramakrishnan Venkitasubramaniam

Presented by Yizhi Huang & Jiaqian Li

2024-02-15

Wait! But the title is. . . ?

Wait! But the title is. . . ?

Helmstedt, Holy Roman Empire, 1799.

Wait! But the title is. . . ?

Helmstedt, Holy Roman Empire, 1799.

Carl Friedrich Gauss

Wait! But the title is. . . ?

Helmstedt, Holy Roman Empire, 1799.

Carl Friedrich Gauss

Johann Friedrich Pfaff

Wait! But the title is. . . ?

Helmstedt, Holy Roman Empire, 1799.

Carl Friedrich Gauss

Johann Friedrich Pfaff

- Trying to embarrass Gauss, Pfaff gives Gauss a hard proposition x, and asks him to either provide a proof w for x, or claim x is false.

Wait! But the title is. . . ?

Helmstedt, Holy Roman Empire, 1799.

Carl Friedrich Gauss

Johann Friedrich Pfaff

- Trying to embarrass Gauss, Pfaff gives Gauss a hard proposition x, and asks him to either provide a proof w for x, or claim x is false.
- If Gauss claims x is false, no way for Pfaff to verify!

Wait! But the title is. . . ?

Helmstedt, Holy Roman Empire, 1799.

Carl Friedrich Gauss

Johann Friedrich Pfaff

- Trying to embarrass Gauss, Pfaff gives Gauss a hard proposition x, and asks him to either provide a proof w for x, or claim x is false.
- If Gauss claims x is false, no way for Pfaff to verify!
- What if Pfaff always gives Gauss a true statement so that he can verify Gauss' solution? Does this makes the task easier for Gauss?

Wait! But the title is. . . ?

Helmstedt, Holy Roman Empire, 1799.

Carl Friedrich Gauss

Johann Friedrich Pfaff

- Trying to embarrass Gauss, Pfaff gives Gauss a hard proposition x, and asks him to either provide a proof w for x, or claim x is false.
- If Gauss claims x is false, no way for Pfaff to verify!
- What if Pfaff always gives Gauss a true statement so that he can verify Gauss' solution? Does this makes the task easier for Gauss?
- This gives a *promise-true* **NP** search problem.

Wait! But the title is. . . ?

Helmstedt, Holy Roman Empire, 1799.

Carl Friedrich Gauss

Johann Friedrich Pfaff

- Trying to embarrass Gauss, Pfaff gives Gauss a hard proposition x, and asks him to either provide a proof w for x, or claim x is false.
- If Gauss claims x is false, no way for Pfaff to verify!
- What if Pfaff always gives Gauss a true statement so that he can verify Gauss' solution? Does this makes the task easier for Gauss?
- This gives a *promise-true* **NP** search problem.
- So the question is: are promise-true **NP** search problems easier than **NP** search problems?

Wait! But the title is. . . ?

Helmstedt, Holy Roman Empire, 1799.

Carl Friedrich Gauss

Johann Friedrich Pfaff

- The question is: are promise-true **NP** search problems easier than **NP** search problems?
- This paper proved that hard-on-average **NP** problems imply OWF or hard-on-average **TFNP** problems.

Wait! But the title is. . . ?

Helmstedt, Holy Roman Empire, 1799.

Carl Friedrich Gauss

Johann Friedrich Pfaff

- The question is: are promise-true **NP** search problems easier than **NP** search problems?
- This paper proved that hard-on-average **NP** problems imply OWF or hard-on-average **TFNP** problems.
- Both inverting OWF and **TFNP** are promise-true!

Wait! But the title is. . . ?

Helmstedt, Holy Roman Empire, 1799.

Carl Friedrich Gauss

Johann Friedrich Pfaff

- The question is: are promise-true **NP** search problems easier than **NP** search problems?
- This paper proved that hard-on-average **NP** problems imply OWF or hard-on-average **TFNP** problems.
- Both inverting OWF and **TFNP** are promise-true!
- Therefore—

TL;DR

NO.