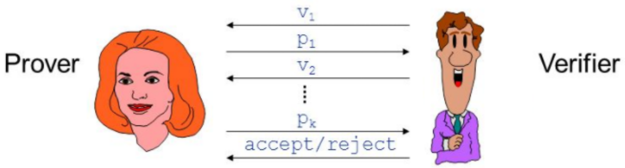
On the power of Statistical Zero Knowledge

Lijie Chen
Joint work with Adam Bouland, Dhiraj Holden,
Justin Thaler and Prashant Nalini Vasudevan
Most graphics are credited to Adam Bouland

UC Berkeley
MIT
Georgetown University

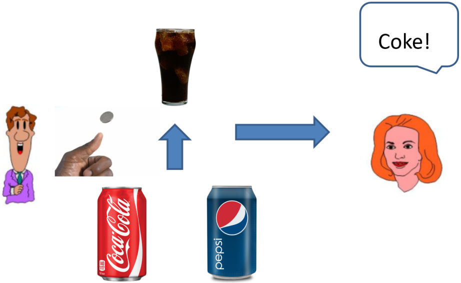
October 17, 2017

Zero Knowledge Proof **[Goldwasser Micali Rackoff '84]**



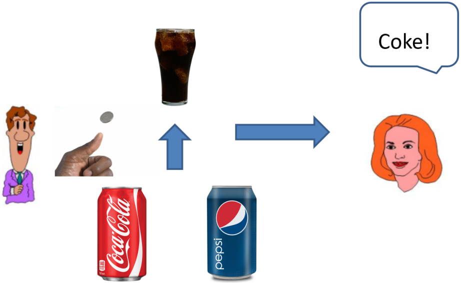
- Alice wants to convince Bob that a certain statement is true,
  - but doesn't want him to know anything more.

Example : Coke-Pepsi Challenge



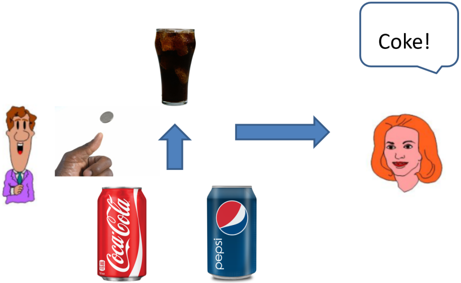
- Alice wants to convince Bob that coke and pepsi are different.

Example : Coke-Pepsi Challenge



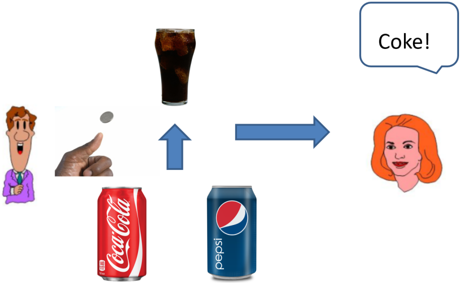
- Alice wants to convince Bob that coke and pepsi are different.
- **Protocol**: Bob flips a random coin, secretly pours coke or pepsi into a glass.

Example : Coke-Pepsi Challenge



- Alice wants to convince Bob that coke and pepsi are different.
- **Protocol**: Bob flips a random coin, secretly pours coke or pepsi into a glass.
- Alice answers whether it is coke or pepsi.

Example : Coke-Pepsi Challenge



- Alice wants to convince Bob that coke and pepsi are different.
- **Protocol**: Bob flips a random coin, secretly pours coke or pepsi into a glass.
- Alice answers whether it is coke or pepsi.
- **Zero knowledge**: since Bob already knew the answer.

Zero Knowledge Proof : Formal Definition

- **Bob doesn't know any additional information**:

Zero Knowledge Proof : Formal Definition

- **Bob doesn't know any additional information**:
- $\Leftrightarrow$ Everything Bob learns from Alice, he can produce by himself.

Zero Knowledge Proof : Formal Definition

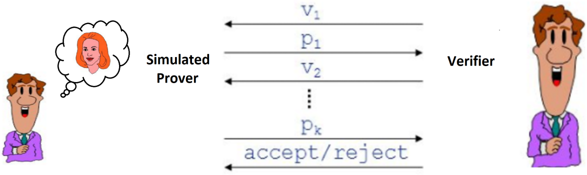
- **Bob doesn't know any additional information**:
- $\Leftrightarrow$ Everything Bob learns from Alice, he can produce by himself.
- All information Bob gets from Alice is a (distribution of) conversation which convinced him.

Zero Knowledge Proof : Formal Definition

- **Bob doesn't know any additional information**:
- $\Leftrightarrow$ Everything Bob learns from Alice, he can produce by himself.
- All information Bob gets from Alice is a (distribution of) conversation which convinced him.
- $\Pi_{A \leftrightarrow B}$: the distribution of the conversation between Alice and Bob.

Zero Knowledge Proof : Formal Definition

- **Bob doesn't know any additional information**:
- $\Leftrightarrow$ Everything Bob learns from Alice, he can produce by himself.
- All information Bob gets from Alice is a (distribution of) conversation which convinced him.
- $\Pi_{A \leftrightarrow B}$: the distribution of the conversation between Alice and Bob.
- $\Leftrightarrow$ Bob can produce a distribution of the conversation Π_B which "looks like" $\Pi_{A \leftrightarrow B}$. (In the YES case.)



Statistical Zero Knowledge Proof (SZK)

- By $\Pi_{A \leftrightarrow B}$ "looks like" Π_B, in SZK, it means...

Statistical Zero Knowledge Proof (SZK)

- By $\Pi_{A \leftrightarrow B}$ "looks like" Π_B, in SZK, it means...
- (**Statistical Zero Knowledge Proof**) SZK : Roughly the same, the total variational distance between $\Pi_{A \leftrightarrow B}$ and Π_B are inverse exponentially small. (In the YES case)

Statistical Zero Knowledge Proof (SZK)

- By $\Pi_{A \leftrightarrow B}$ "looks like" Π_B, in SZK, it means...
- (**Statistical Zero Knowledge Proof**) SZK : Roughly the same, the total variational distance between $\Pi_{A \leftrightarrow B}$ and Π_B are inverse exponentially small. (In the YES case)
- Indeed, our results apply for the following sub-class of SZK.

Statistical Zero Knowledge Proof (SZK)

- By $\Pi_{A \leftrightarrow B}$ "looks like" Π_B, in SZK, it means...
- (**Statistical Zero Knowledge Proof**) SZK : Roughly the same, the total variational distance between $\Pi_{A \leftrightarrow B}$ and Π_B are inverse exponentially small. (In the YES case)
- Indeed, our results apply for the following sub-class of SZK.
- (**Non-Interactive Statistical Zero Knowledge Proof**) NISZK : Alice doesn't interact with Bob, just say something and leave (they share public random bits)

This work: Exploring the Power of SZK
Motivation

- Evidence that SZK contains some very hard problems.

This work: Exploring the Power of SZK
Motivation

- Evidence that SZK contains some very hard problems.
- Relationship between several different kinds of proof systems related to SZK.

Our Results

- Result I : Query SZK is very powerful.
  - Black-box SZK contains problem outside of PP, open since **[Watrous'02]**. (an oracle separation between SZK and PP)

Our Results

- Result I : Query SZK is very powerful.
  - Black-box SZK contains problem outside of PP, open since **[Watrous'02]**. (an oracle separation between SZK and PP)
- Result II : Communication SZK is very powerful.
  - SZK^{cc} lies outside of UPP^{cc}, open since **[Göös, Pitassi and Watson'15]**.

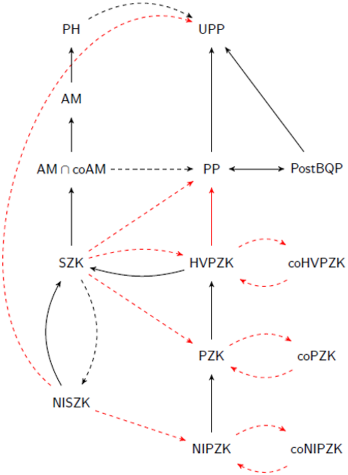
Our Results

- Result I : Query SZK is very powerful.
  - Black-box SZK contains problem outside of PP, open since **[Watrous'02]**. (an oracle separation between SZK and PP)
- Result II : Communication SZK is very powerful.
  - $\mathrm{SZK}^{\mathrm{cc}}$ lies outside of $\mathrm{UPP}^{\mathrm{cc}}$, open since **[Göös, Pitassi and Watson'15]**.
- Result III : SZK may be larger than PZK.
  - Black-box SZK contains problems outside of PZK, open since **[Aiello Hastad'91]**. (an oracle separation between SZK and PZK).

Our Results

- Result I : Query SZK is very powerful.
  - Black-box SZK contains problem outside of PP, open since **[Watrous'02]**. (an oracle separation between SZK and PP)
- Result II : Communication SZK is very powerful.
  - $\mathrm{SZK}^{\mathrm{cc}}$ lies outside of $\mathrm{UPP}^{\mathrm{cc}}$, open since **[Göös, Pitassi and Watson'15]**.
- Result III : SZK may be larger than PZK.
  - Black-box SZK contains problems outside of PZK, open since **[Aiello Hastad'91]**. (an oracle separation between SZK and PZK).
- And more!

New Oracle Separations (Result I & III)



solid line : containment

dashed line : separation

black : known results

red : new results

Result I : Query SZK is very powerful

- Applications to Crypto $\Rightarrow$ need SZK to contain problems outside of P or BPP.
    - Quadratic Residuosity.
    - Some lattice problems.

Result I : Query SZK is very powerful

- Applications to Crypto $\Rightarrow$ need SZK to contain problems outside of P or BPP.
  - Quadratic Residuosity.
  - Some lattice problems.
- What is the evidence that SZK contains some really hard problems?

Result I : Query SZK is very powerful

- Applications to Crypto $\Rightarrow$ need SZK to contain problems outside of P or BPP.
  - Quadratic Residuosity.
  - Some lattice problems.
- What is the evidence that SZK contains some really hard problems?
- **Obstacle:** $\mathsf{P} \neq \mathsf{SZK}$ implies $\mathsf{P} \neq \mathsf{NP}$
  - $\mathsf{P} = \mathsf{NP} \implies \mathsf{P} = \mathsf{PH}$ and $\mathsf{SZK} \subseteq \mathsf{PH}$.

Result I : Query SZK is very powerful

- So what about the relativized(query) version of SZK (e.g. oracle separation?)
  - **Query Complexity**: just count the number of queries to an oracle, and don't have limitation on computational resources.

Result I : Query SZK is very powerful

- So what about the relativized(query) version of SZK (e.g. oracle separation?)
  - **Query Complexity**: just count the number of queries to an oracle, and don't have limitation on computational resources.
- Relativized SZK contains problems outside of:

Result I : Query SZK is very powerful

- So what about the relativized(query) version of SZK (e.g. oracle separation?)
  - **Query Complexity**: just count the number of queries to an oracle, and don't have limitation on computational resources.
- Relativized SZK contains problems outside of:
  - **[Aiello Hastad'91]**: BPP

Result I : Query SZK is very powerful

- So what about the relativized(query) version of SZK (e.g. oracle separation?)
  - **Query Complexity**: just count the number of queries to an oracle, and don't have limitation on computational resources.

- Relativized SZK contains problems outside of:
  - **[Aiello Hastad'91]**: BPP
  - **[Aaronson'02]**: BQP

Result I : Query SZK is very powerful

- So what about the relativized(query) version of SZK (e.g. oracle separation?)
  - **Query Complexity**: just count the number of queries to an oracle, and don't have limitation on computational resources.
- Relativized SZK contains problems outside of:
  - **[Aiello Hastad'91]**: BPP
  - **[Aaronson'02]**: BQP
  - **[Aaronson'12]**: QMA (quantum version of NP)

Result I : Query SZK is very powerful

- So what about the relativized(query) version of SZK (e.g. oracle separation?)
  - **Query Complexity**: just count the number of queries to an oracle, and don't have limitation on computational resources.

- Relativized SZK contains problems outside of:
  - **[Aiello Hastad'91]**: BPP
  - **[Aaronson'02]**: BQP
  - **[Aaronson'12]**: QMA (quantum version of NP)

- **[Watrous'02]**: Does relativized SZK contain problems outside of PP? (PP is the smallest natural classical class containing BQP.)

Probabilistic Polynomial-Time (PP)

- Languages decidable by poly-time randomized algorithms with unbounded error.
  - If Yes: $\Pr[\text{accept}] > 1/2$.
  - If No: $\Pr[\text{accept}] < 1/2$.
  - Gap may be exponentially small. (because there is only polynomial number of coin flips).
- PP is very powerful : PP contains NP and P^{PP} contains PH by **[Toda'91]**.

PP in query complexity

- A PP algorithm in query complexity is similar to randomized query algorithms, except for that it only needs to be correct on every input w.p. > 0.5.

PP in query complexity

- A PP algorithm in query complexity is similar to randomized query algorithms, except for that it only needs to be correct on every input w.p. > 0.5.
    - The complexity of an algorithm is the **sum** of the number of bits it queried

PP in query complexity

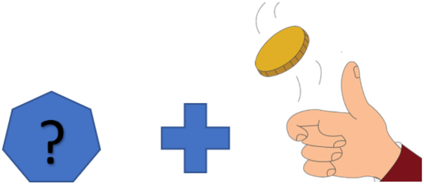
- A PP algorithm in query complexity is similar to randomized query algorithms, except for that it only needs to be correct on every input w.p. > 0.5.
  - The complexity of an algorithm is the **sum** of the number of bits it queried
  - and the number of random bits it used.

PP in query complexity

- A PP algorithm in query complexity is similar to randomized query algorithms, except for that it only needs to be correct on every input w.p. > 0.5.
  - The complexity of an algorithm is the **sum** of the number of bits it queried
  - and the number of random bits it used.
  - a d-cost PP query algorithm must have gap $\geq 2^{-d}$.

PP in query complexity

- A PP algorithm in query complexity is similar to randomized query algorithms, except for that it only needs to be correct on every input w.p. > 0.5.
  - The complexity of an algorithm is the **sum** of the number of bits it queried
  - and the number of random bits it used.
  - a d-cost PP query algorithm must have gap $\geq 2^{-d}$.



UPP (Unrestricted Probabilistic Polynomial-Time) in query complexity

- similar to PP query algorithms.

UPP (Unrestricted Probabilistic Polynomial-Time) in query complexity

- similar to PP query algorithms.
- an UPP algorithm in query complexity is **not** charged for using random bits (or runtime).
- only charged for query.

UPP (Unrestricted Probabilistic Polynomial-Time) in query complexity

- similar to PP query algorithms.
- an UPP algorithm in query complexity is **not** charged for using random bits (or runtime).
- only charged for query.
- the gap can be arbitrarily small.

UPP (Unrestricted Probabilistic Polynomial-Time) in query complexity

- similar to PP query algorithms.
- an UPP algorithm in query complexity is **not** charged for using random bits (or runtime).
- only charged for query.
- the gap can be arbitrarily small.
- UPP query complexity is equivalent to
    - **Threshold Degree of f**: $\deg_{\pm}(f)$, the least degree polynomial p which sign-represents f
    - $p(x) > 0$ when $f(x) = 1$, and $p(x) < 0$ when $f(x) = 0$.

Result I : Query SZK is very powerful

- **Result I**: relativized version of SZK (indeed NISZK) contains problem outside of PP (even UPP).

Result I : Query SZK is very powerful

- **Result I**: relativized version of SZK (indeed NISZK) contains problem outside of PP (even UPP).
- A query problem with $\mathrm{polylog}(n)$-SZK algorithm, has no $o(n^{1/4})$ UPP algorithm.
  - implies an oracle separation between SZK and PP. (Answer **[Watrous'02]**).

Result I : Query SZK is very powerful

- **Result I**: relativized version of SZK (indeed NISZK) contains problem outside of PP (even UPP).
- A query problem with $\mathrm{polylog}(n)$-SZK algorithm, has no $o(n^{1/4})$ UPP algorithm.
  - implies an oracle separation between SZK and PP. (Answer **[Watrous'02]**).
- since PP = PostBQP (**[Aaronson'05]**), even **post-selected quantum algorithms** can not crack SZK in a black-box way.

Result I : Query SZK is very powerful

- **Result I**: relativized version of SZK (indeed NISZK) contains problem outside of PP (even UPP).
- A query problem with $\mathrm{polylog}(n)$-SZK algorithm, has no $o(n^{1/4})$ UPP algorithm.
  - implies an oracle separation between SZK and PP. (Answer **[Watrous'02]**).
- since PP $=$ PostBQP (**[Aaronson'05]**), even **post-selected quantum algorithms** can not crack SZK in a black-box way.
- A brief overview of how is it proved.

Previous Works and Difficulty

- Difficulty: All previous hard problems from SZK are actually in PP.

Previous Works and Difficulty

- Difficulty: All previous hard problems from SZK are actually in PP.
- Collision: Distinguish whether a given function from $[n]$ to $[n]$ is 1-to-1 or 2-to-1.

Previous Works and Difficulty

- Difficulty: All previous hard problems from SZK are actually in PP.
- Collision: Distinguish whether a given function from $[n]$ to $[n]$ is 1-to-1 or 2-to-1.
  - has a constant query SZK protocol.

Previous Works and Difficulty

- Difficulty: All previous hard problems from SZK are actually in PP.
- Collision: Distinguish whether a given function from $[n]$ to $[n]$ is 1-to-1 or 2-to-1.
  - has a constant query SZK protocol.
  - requires $\Omega(n^{1/3})$ (bounded) approximate polynomial degree. **[Aaronson'02],[Aaronson and Shi'04],[Ambainis'05],[Kutin'05]**
    - which implies the $\Omega(n^{1/3})$ quantum query complexity lower bound.

Previous Works and Difficulty

- Difficulty: All previous hard problems from SZK are actually in PP.
- Collision: Distinguish whether a given function from $[n]$ to $[n]$ is 1-to-1 or 2-to-1.
  - has a constant query SZK protocol.
  - requires $\Omega(n^{1/3})$ (bounded) approximate polynomial degree. **[Aaronson'02],[Aaronson and Shi'04],[Ambainis'05],[Kutin'05]**
    - which implies the $\Omega(n^{1/3})$ quantum query complexity lower bound.
- Unfortunately it is in PP:
  - whether there are collisions, in fact in NP.

Previous Works and Difficulty

- Difficulty: All previous hard problems from SZK are actually in PP.
- Collision: Distinguish whether a given function from $[n]$ to $[n]$ is 1-to-1 or 2-to-1.
  - has a constant query SZK protocol.
  - requires $\Omega(n^{1/3})$ (bounded) approximate polynomial degree. **[Aaronson'02],[Aaronson and Shi'04],[Ambainis'05],[Kutin'05]**
    - which implies the $\Omega(n^{1/3})$ quantum query complexity lower bound.
- Unfortunately it is in PP:
  - whether there are collisions, in fact in NP.
- **Sad reality**: PP is too powerful.

Intuition: What is the Weakness of PP?

- Hand-waving Intuition: find something which is **easy** for SZK, but **hard** for PP. (In query complexity setting)

Intuition: What is the Weakness of PP?

- Hand-waving Intuition: find something which is **easy** for SZK, but **hard** for PP. (In query complexity setting)
- **Randomized Reduction**! (the BP operator).
  - $\mathrm{BP} \cdot \mathcal{C}$: $L \in \mathrm{BP} \cdot \mathcal{C}$ iff there is a poly-time randomized reduction T and a language $L' \in \mathcal{C}$ such that

$$\begin{aligned} x \in L &\implies \Pr[T(x) \in L'] \geq 2/3 \\ x \notin L &\implies \Pr[T(x) \in L'] \leq 1/3 \end{aligned}$$

Intuition: What is the Weakness of PP?

- Hand-waving Intuition: find something which is **easy** for SZK, but **hard** for PP. (In query complexity setting)
- **Randomized Reduction**! (the BP operator).
  - $\mathrm{BP} \cdot \mathcal{C}$: $L \in \mathrm{BP} \cdot \mathcal{C}$ iff there is a poly-time randomized reduction T and a language $L' \in \mathcal{C}$ such that

$$x \in L \implies \Pr[T(x) \in L'] \geq 2/3$$
$$x \notin L \implies \Pr[T(x) \in L'] \leq 1/3$$

  - $\mathrm{BP} \cdot \mathrm{NP} = \mathrm{AM}$, $\mathrm{BP} \cdot \mathrm{P} = \mathrm{BPP}$.

Intuition: What is the Weakness of PP?

- Hand-waving Intuition: find something which is **easy** for SZK, but **hard** for PP. (In query complexity setting)
- **Randomized Reduction**! (the BP operator).
  - $\mathsf{BP} \cdot \mathcal{C}$: $L \in \mathsf{BP} \cdot \mathcal{C}$ iff there is a poly-time randomized reduction T and a language $L' \in \mathcal{C}$ such that

$$x \in L \implies \Pr[T(x) \in L'] \geq 2/3$$
$$x \notin L \implies \Pr[T(x) \in L'] \leq 1/3$$

  - $\mathsf{BP} \cdot \mathsf{NP} = \mathsf{AM}$, $\mathsf{BP} \cdot \mathsf{P} = \mathsf{BPP}$.
- **Easy for** SZK: SZK is closed under-randomized reduction. ($\mathsf{BP} \cdot \mathsf{SZK} = \mathsf{SZK}$ relative to all oracles). **[Sahai and Vadhan'97]**

Intuition: What is the Weakness of PP?

- Hand-waving Intuition: find something which is **easy** for SZK, but **hard** for PP. (In query complexity setting)
- **Randomized Reduction**! (the BP operator).
  - $\mathsf{BP} \cdot \mathcal{C}$: $L \in \mathsf{BP} \cdot \mathcal{C}$ iff there is a poly-time randomized reduction T and a language $L' \in \mathcal{C}$ such that

$$\begin{aligned} x \in L &\implies \Pr[T(x) \in L'] \geq 2/3 \\ x \notin L &\implies \Pr[T(x) \in L'] \leq 1/3 \end{aligned}$$

  - $\mathsf{BP} \cdot \mathsf{NP} = \mathsf{AM}$, $\mathsf{BP} \cdot \mathsf{P} = \mathsf{BPP}$.
- **Easy for** SZK: SZK is closed under-randomized reduction. ($\mathsf{BP} \cdot \mathsf{SZK} = \mathsf{SZK}$ relative to all oracles). **[Sahai and Vadhan'97]**
- **Hard for** PP: PP is not closed under randomized reduction for some oracle $\mathcal{O}$.
  - In fact, $(\mathsf{BP} \cdot \mathsf{NP})^{\mathcal{O}} = \mathsf{AM}^{\mathcal{O}} \not\subset \mathsf{PP}^{\mathcal{O}}$ **[Vereshchagin'92]**.

What is randomized reduction in query complexity?

- What we have : a function $f\colon \{0,1\}^M \to \{0,1\}$.

What is randomized reduction in query complexity?

- What we have : a function $f \colon \{0,1\}^M \to \{0,1\}$.
- **Gapped Majority:** $F := \mathsf{GapMaj}_d(f) : \{0,1\}^{d \cdot M} \to \{0,1\}$

What is randomized reduction in query complexity?

- What we have : a function $f\colon \{0,1\}^M \to \{0,1\}$.
- **Gapped Majority:** $F := \mathsf{GapMaj}_d(f) : \{0,1\}^{d \cdot M} \to \{0,1\}$
  - Given d copies of inputs $x_1, x_2, \ldots, x_d$ to f.
  - $x = (x_1, x_2, \ldots, x_d)$.

What is randomized reduction in query complexity?

- What we have : a function $f\colon \{0,1\}^M \to \{0,1\}$.
- **Gapped Majority:** $F := \mathsf{GapMaj}_d(f) : \{0,1\}^{d \cdot M} \to \{0,1\}$
  - Given d copies of inputs $x_1, x_2, \ldots, x_d$ to f.
  - $x = (x_1, x_2, \ldots, x_d)$.
  - $F(x) = 1$ when $2/3$ of the $f(x_i)$'s are 1.

What is randomized reduction in query complexity?

- What we have : a function $f\colon \{0,1\}^M \to \{0,1\}$.
- **Gapped Majority:** $F := \mathsf{GapMaj}_d(f) : \{0,1\}^{d \cdot M} \to \{0,1\}$
  - Given d copies of inputs $x_1, x_2, \ldots, x_d$ to f.
  - $x = (x_1, x_2, \ldots, x_d)$.
  - $F(x) = 1$ when $2/3$ of the $f(x_i)$'s are 1.
  - $F(x) = 0$ when $2/3$ of the $f(x_i)$'s are 0.

What is randomized reduction in query complexity?

- What we have : a function $f\colon \{0,1\}^M \to \{0,1\}$.
- **Gapped Majority:** $F := \mathsf{GapMaj}_d(f) : \{0,1\}^{d \cdot M} \to \{0,1\}$
  - Given d copies of inputs $x_1, x_2, \ldots, x_d$ to f.
  - $x = (x_1, x_2, \ldots, x_d)$.
  - $F(x) = 1$ when $2/3$ of the $f(x_i)$'s are 1.
  - $F(x) = 0$ when $2/3$ of the $f(x_i)$'s are 0.
  - undefined otherwise.

What is randomized reduction in query complexity?

- What we have : a function $f\colon \{0,1\}^M \to \{0,1\}$.
- **Gapped Majority:** $F := \mathrm{GapMaj}_d(f) : \{0,1\}^{d \cdot M} \to \{0,1\}$
  - Given d copies of inputs $x_1, x_2, \ldots, x_d$ to f.
  - $x = (x_1, x_2, \ldots, x_d)$.
  - $F(x) = 1$ when $2/3$ of the $f(x_i)$'s are 1.
  - $F(x) = 0$ when $2/3$ of the $f(x_i)$'s are 0.
  - undefined otherwise.
- **Captures what can be randomized reduced to f.**

What is randomized reduction in query complexity?

- What we have : a function $f\colon \{0,1\}^M \to \{0,1\}$.
- **Gapped Majority:** $F := \mathrm{GapMaj}_d(f) : \{0,1\}^{d \cdot M} \to \{0,1\}$
  - Given d copies of inputs $x_1, x_2, \ldots, x_d$ to f.
  - $x = (x_1, x_2, \ldots, x_d)$.
  - $F(x) = 1$ when $2/3$ of the $f(x_i)$'s are 1.
  - $F(x) = 0$ when $2/3$ of the $f(x_i)$'s are 0.
  - undefined otherwise.
- **Captures what can be randomized reduced to f.**
- **Intuition**:
  - Since randomized reduction is hard for PP, $\mathrm{GapMaj}_d(f)$ should be harder than f for PP in some sense.

Core Technique Result: Hardness Amplification Theorem Gapped Majority is really hard for PP

f : requires a degree d poly to approximate within L_∞ distance 0.1

|f(x) – p(x)| <= 0.1

→

F(GapMaj(f)) : requires a degree $\Omega(d)$ poly to sign-represents it.

p(x) > 0 if f(x) = 1
p(x) < 0 if f(x) = 0

Composition with Gapped-Majority:
GapMaj(f):
d copies of f on inputs
x_1,x_2,…,x_d
1 when 2/3 of f(x_i)'s are 1
0 when 2/3 of f(x_i)'s are 0

Core Technique Result: Hardness Amplification Theorem
Gapped Majority is really hard for PP

f : requires a degree d poly to approximate within L_∞ distance 0.1

|f(x) – p(x)| <= 0.1

F(GapMaj(f)) : requires a degree $\Omega(d)$ poly to sign-represents it.

p(x) > 0 if f(x) = 1
p(x) < 0 if f(x) = 0

Composition with
Gapped-Majority:
GapMaj(f):
d copies of f on inputs
x_1,x_2,…,x_d
1 when 2/3 of f(x_i)'s are 1
0 when 2/3 of f(x_i)'s are 0

- Proved by constructing the dual object to witness the high threshold degree. (**[Sherstov'14]**,**[Bun and Thaler'15]**).

Core Technique Result: Hardness Amplification Theorem
Gapped Majority is really hard for PP

f : requires a degree d poly to approximate within L_∞ distance 0.1

|f(x) – p(x)| <= 0.1

F(GapMaj(f)) : requires a degree $\Omega(d)$ poly to sign-represents it.

p(x) > 0 if f(x) = 1
p(x) < 0 if f(x) = 0

Composition with Gapped-Majority:
GapMaj(f):
d copies of f on inputs
x_1,x_2,…,x_d
1 when 2/3 of f(x_i)'s are 1
0 when 2/3 of f(x_i)'s are 0

- Proved by constructing the dual object to witness the high threshold degree. (**[Sherstov'14]**,**[Bun and Thaler'15]**).
- Actually it has a converse, when f has a degree d L_∞-approximate-polynomial, $\mathrm{GapMaj}_d(f)$ has threshold degree $O(d)$.

SZK$^{\mathcal{O}} \not\subseteq$ UPP$^{\mathcal{O}}$

- Collision : Distinguish whether a given function from $[n]$ to $[n]$ is 1-to-1 or 2-to-1.
    - constant query SZK protocol.
    - require $\Omega(n^{1/3})$ (bounded) approximate polynomial degree. **[Aaronson'02],[Aaronson and Shi'04],[Ambainis'05],[Kutin'05]**

$\mathsf{SZK}^{\mathcal{O}} \not\subseteq \mathsf{UPP}^{\mathcal{O}}$

- Collision : Distinguish whether a given function from $[n]$ to $[n]$ is 1-to-1 or 2-to-1.
  - constant query SZK protocol.
  - require $\Omega(n^{1/3})$ (bounded) approximate polynomial degree. **[Aaronson'02],[Aaronson and Shi'04],[Ambainis'05],[Kutin'05]**
- Compose Gapped-Majority with Collision.
  - $F := \mathsf{GapMaj}_{n^{1/3}}(\mathsf{Collision})$.

$\mathsf{SZK}^{\mathcal{O}} \not\subseteq \mathsf{UPP}^{\mathcal{O}}$

- Collision : Distinguish whether a given function from $[n]$ to $[n]$ is 1-to-1 or 2-to-1.
  - constant query SZK protocol.
  - require $\Omega(n^{1/3})$ (bounded) approximate polynomial degree. **[Aaronson'02],[Aaronson and Shi'04],[Ambainis'05],[Kutin'05]**
- Compose Gapped-Majority with Collision.
  - $F := \mathsf{GapMaj}_{n^{1/3}}(\mathsf{Collision})$.
  - F still in SZK, because $\mathsf{BP} \cdot \mathsf{SZK} = \mathsf{SZK}$ (SZK is closed under randomized reduction). **[Sahai and Vadhan'97]**

$\mathsf{SZK}^{\mathcal{O}} \not\subseteq \mathsf{UPP}^{\mathcal{O}}$

- Collision : Distinguish whether a given function from $[n]$ to $[n]$ is 1-to-1 or 2-to-1.
  - constant query SZK protocol.
  - require $\Omega(n^{1/3})$ (bounded) approximate polynomial degree. **[Aaronson'02],[Aaronson and Shi'04],[Ambainis'05],[Kutin'05]**
- Compose Gapped-Majority with Collision.
  - $F := \mathrm{GapMaj}_{n^{1/3}}(\mathrm{Collision})$.
  - F still in SZK, because $\mathsf{BP} \cdot \mathsf{SZK} = \mathsf{SZK}$ (SZK is closed under randomized reduction). **[Sahai and Vadhan'97]**
  - F has threshold degree $\Omega(n^{1/4})$. **[Our Work]**

$\mathsf{SZK}^{\mathcal{O}} \not\subset \mathsf{UPP}^{\mathcal{O}}$

- Collision : Distinguish whether a given function from $[n]$ to $[n]$ is 1-to-1 or 2-to-1.
  - constant query SZK protocol.
  - require $\Omega(n^{1/3})$ (bounded) approximate polynomial degree. **[Aaronson'02],[Aaronson and Shi'04],[Ambainis'05],[Kutin'05]**
- Compose Gapped-Majority with Collision.
  - $F := \mathsf{GapMaj}_{n^{1/3}}(\mathsf{Collision})$.
  - F still in SZK, because $\mathsf{BP} \cdot \mathsf{SZK} = \mathsf{SZK}$ (SZK is closed under randomized reduction). **[Sahai and Vadhan'97]**
  - F has threshold degree $\Omega(n^{1/4})$. **[Our Work]**
- Implies our separation.

Result II : Communication SZK is very powerful

- **Result 2**: SZK^{cc} (even NISZK^{cc}) is not contained in UPP^{cc}.

Result II : Communication SZK is very powerful

- **Result 2**: $\mathsf{SZK}^{\mathrm{cc}}$ (even $\mathsf{NISZK}^{\mathrm{cc}}$) is not contained in $\mathsf{UPP}^{\mathrm{cc}}$.
- Answers **[Göös, Pitassi and Watson'15]**.
  - **[GPW'15]** : can we show $(\mathsf{AM}^{\mathrm{cc}} \cap \mathsf{coAM}^{\mathrm{cc}}) \not\subseteq \mathsf{UPP}^{\mathrm{cc}}$?
  - $\mathsf{SZK} \subseteq (\mathsf{AM}^{\mathrm{cc}} \cap \mathsf{coAM}^{\mathrm{cc}}) \subseteq \mathsf{AM}^{\mathrm{cc}}$.

Result II : Communication SZK is very powerful

UPP^{CC}

Can prove lower bounds

AM^{CC}

Can't prove lower bounds

Result II : Communication SZK is very powerful

UPP^{CC}
Can prove lower bounds

AM^{CC}
Can't prove lower bounds

- AM^{cc} : Notoriously hard to prove a communication complexity lower bound against it (first step toward proving lower bound for PH^{cc}).

Result II : Communication SZK is very powerful

$\mathrm{UPP}^{\mathrm{CC}}$
Can prove lower bounds

$\mathrm{AM}^{\mathrm{CC}}$
Can't prove lower bounds

- $\mathrm{AM}^{\mathrm{cc}}$: Notoriously hard to prove a communication complexity lower bound against it (first step toward proving lower bound for $\mathrm{PH}^{\mathrm{cc}}$).
- $\mathrm{UPP}^{\mathrm{cc}}$: the **strongest class** we know how to prove non-trivial communication lower bound.

Result II : Communication SZK is very powerful

UPP^{CC}

Can prove lower bounds

$\not\supseteq$

AM^{CC} NISZK^{CC}

Can't prove lower bounds

Result II : Communication SZK is very powerful

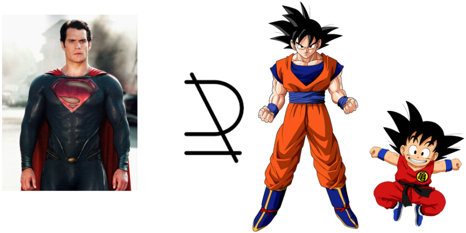
$\not\supseteq$

UPP^{CC} — *Can* prove lower bounds

AM^{CC} NISZK^{CC} — *Can't* prove lower bounds

- Not possible to use UPP lower bound to prove AM^{cc} lower bound.

Result II : Communication SZK is very powerful

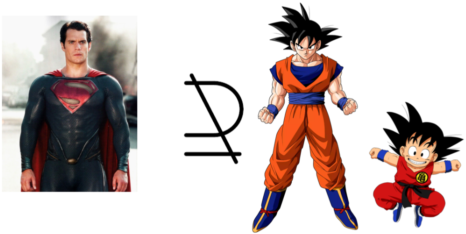


$\mathrm{UPP}^{\mathrm{CC}}$ $\not\supseteq$ $\mathrm{AM}^{\mathrm{CC}}$ $\mathrm{NISZK}^{\mathrm{CC}}$

Can prove lower bounds *Can't* prove lower bounds

- Not possible to use UPP lower bound to prove AM^{cc} lower bound.
- Some related previous work:

Result II : Communication SZK is very powerful



UPP^{CC} — *Can* prove lower bounds

AM^{CC} NISZK^{CC} — *Can't* prove lower bounds

- Not possible to use UPP lower bound to prove AM^{cc} lower bound.
- Some related previous work:
  - **[Razbarov and Sherstov'2010]** : $\text{PH}^{\text{cc}} \not\subseteq \text{UPP}^{\text{cc}}$ (infact Σ_2^{cc}, $\text{AM}^{\text{cc}} \subseteq \Sigma_2^{\text{cc}}$).

Result II : Communication SZK is very powerful

$\not\supseteq$

UPP^{CC} *Can* prove lower bounds

AM^{CC} NISZK^{CC} *Can't* prove lower bounds

- Not possible to use UPP lower bound to prove AM^{cc} lower bound.
- Some related previous work:
    - **[Razbarov and Sherstov'2010]** : PH^{cc} $\not\subseteq$ UPP^{cc} (infact Σ_2^{cc}, AM^{cc} $\subseteq \Sigma_2^{cc}$).
    - **[Klauck'2011]**: (AM^{cc} $\cap$ coAM^{cc}) $\not\subseteq$ PP^{cc}.

Result II : Communication SZK is very powerful

$\not\supseteq$

UPP^{CC} *Can* prove lower bounds

AM^{CC} NISZK^{CC} *Can't* prove lower bounds

- Not possible to use UPP lower bound to prove AM^{cc} lower bound.
- Some related previous work:
  - **[Razbarov and Sherstov'2010]** : $\text{PH}^{\text{cc}} \not\subseteq \text{UPP}^{\text{cc}}$ (infact Σ_2^{cc}, $\text{AM}^{\text{cc}} \subseteq \Sigma_2^{\text{cc}}$).
  - **[Klauck'2011]**: $(\text{AM}^{\text{cc}} \cap \text{coAM}^{\text{cc}}) \not\subseteq \text{PP}^{\text{cc}}$.
  - **Our improvement** : $\text{NISZK}^{\text{cc}} \not\subseteq \text{UPP}^{\text{cc}}$, $\text{NISZK}^{\text{cc}} \subseteq \text{SZK}^{\text{cc}} \subseteq \text{AM}^{\text{cc}}$.

Result II : Communication SZK is very powerful

- **Moral** : Communication SZK contains some very hard problems(even outside of UPP), which explains why we can't prove lower bounds for $\mathrm{AM}^{\mathrm{cc}}$.

Result III : SZK may be more powerful than PZK
SZK and its friends

- Zero Knowledge : Bob gets no additional information from Alice $\Leftrightarrow$ Bob can produce a “simulated” prover which looks like Alice.

Result III : SZK may be more powerful than PZK
SZK and its friends

- Zero Knowledge : Bob gets no additional information from Alice $\Leftrightarrow$ Bob can produce a “simulated” prover which looks like Alice.
- **Statistical** Zero Knowledge (SZK) : the simulated prover looks the same as Alice except for an inverse exponential total variational distance.

Result III : SZK may be more powerful than PZK
SZK and its friends

- Zero Knowledge : Bob gets no additional information from Alice $\Leftrightarrow$ Bob can produce a "simulated" prover which looks like Alice.

- **Statistical** Zero Knowledge (SZK) : the simulated prover looks the same as Alice except for an inverse exponential total variational distance.

- **Perfect** Zero Knowledge (PZK) : the simulated prover looks exactly the same as Alice.

SZK and its friends

- Zero Knowledge : Bob gets no additional information from Alice $\Leftrightarrow$ Bob can produce a "simulated" prover which looks like Alice.
- **Statistical** Zero Knowledge (SZK) : the simulated prover looks the same as Alice except for an inverse exponential total variational distance.
- **Perfect** Zero Knowledge (PZK) : the simulated prover looks exactly the same as Alice.
- **Non-Interactive** Zero Knowledge (NISZK or NIPZK) : no interaction, Alice says something and just leave. (they share some public random bits).

What is the relationship between these classes?

- Two intriguing open questions here:

What is the relationship between these classes?

- Two intriguing open questions here:
  - Is SZK equal to PZK (or at least an **oracle separation**)? **[Aiello Hastad'91]**

What is the relationship between these classes?

- Two intriguing open questions here:
  - Is SZK equal to PZK (or at least an **oracle separation**)? **[Aiello Hastad'91]**
  - Is PZK closed under complement, the same way that SZK is **[Sahai Vadhan'99]** (or at least an **oracle separation**)?

Our Result

- **Result III**: There exists an oracle $\mathcal{O}$ such that
  - $\mathsf{SZK}^{\mathcal{O}} \neq \mathsf{PZK}^{\mathcal{O}}$.

Our Result

- **Result III**: There exists an oracle $\mathcal{O}$ such that
  - $\mathsf{SZK}^{\mathcal{O}} \neq \mathsf{PZK}^{\mathcal{O}}$.
- We also have
  - $\mathsf{coPZK}^{\mathcal{O}} \neq \mathsf{PZK}^{\mathcal{O}}$.
  - $\mathsf{coNIPZK}^{\mathcal{O}} \neq \mathsf{NIPZK}^{\mathcal{O}}$.

Our Result

- **Result III**: There exists an oracle $\mathcal{O}$ such that
  - $\mathsf{SZK}^{\mathcal{O}} \neq \mathsf{PZK}^{\mathcal{O}}$.
- We also have
  - $\mathsf{coPZK}^{\mathcal{O}} \neq \mathsf{PZK}^{\mathcal{O}}$.
  - $\mathsf{coNIPZK}^{\mathcal{O}} \neq \mathsf{NIPZK}^{\mathcal{O}}$.
- Therefore SZK may be more powerful than PZK, and any proof that SZK = PZK, or PZK = coPZK, must be nonrelativizing.

Technique

- **Lemma**: $\mathsf{PZK}^{\mathcal{O}} \subseteq \mathsf{PP}^{\mathcal{O}}$, relative to all oracle $\mathcal{O}$.

Technique

- **Lemma**: $\mathsf{PZK}^{\mathcal{O}} \subseteq \mathsf{PP}^{\mathcal{O}}$, relative to all oracle $\mathcal{O}$.
  - $\mathsf{SZK}^{\mathcal{O}} \not\subseteq \mathsf{PP}^{\mathcal{O}} \implies \mathsf{SZK}^{\mathcal{O}} \neq \mathsf{PZK}^{\mathcal{O}}$.

Technique

- **Lemma**: $\mathsf{PZK}^{\mathcal{O}} \subseteq \mathsf{PP}^{\mathcal{O}}$, relative to all oracle $\mathcal{O}$.
  - $\mathsf{SZK}^{\mathcal{O}} \not\subseteq \mathsf{PP}^{\mathcal{O}} \implies \mathsf{SZK}^{\mathcal{O}} \neq \mathsf{PZK}^{\mathcal{O}}$.
- For $\mathsf{PZK}^{\mathcal{O}} \neq \mathsf{coPZK}^{\mathcal{O}}$, we use a different proof with another hardness amplification theorem.

Thanks!