Delegating Computation
Via Non-Signaling Strategies

Yael Tauman Kalai

Microsoft Research

and

MIT

Thanks to Justin Holmgren for some slides!
Privacy?

Integrity?
Delegating Computation

Efficiency:
Verifying takes time $\ll T$

Non-Signaling Strategies
Evolution of Proofs
Classical Proofs

$NP = \{\text{languages: membership can be efficiently verified via classical proofs}\}$

$NP \equiv P$
Many functions computable in time $T$ do not have a classical proof verifiable in time $\ll T$.
Delegating Computation

Efficiency:
Verifying takes time $\ll T$
Interactive Proofs

[Goldwasser-Micali-Rackoff85]

motivated by cryptography

Soundness is only required to hold with high probability.
Interactive Proofs are More Powerful!
[Lund-Fortnow-Karloff-Nissan90, Shamir90]

Example: Chess

Any computation can be verified in time $\approx$ space
Multi-Prover Interactive Proofs

[BenOr-Goldwasser-Kilian-Wigderson88]

motivated by cryptography

∀f computable in time $T$:
2-provers can convince verifier that $f(x) = y$, where the runtime of the verifier is only $|x| \cdot \text{polylog}(T)$ and the communication is $\text{polylog}(T)$
Delegating Computation

Efficiency:

Runtime of the verifier $|x| \cdot \text{polylog}(T)$

Runtime of prover $\text{poly}(T)$

Communication complexity $= \text{polylog}(T)$
Classical proofs

Interactive proofs

Multi-prover interactive proofs
Delegating Computation via Interactive Proofs

Delegating Computation via Interactive Proofs

\[ \text{proof: } f(x) \overset{T}{=} y \]

Inefficient!
Runs in time \(2^{S^2}\)

Runs in time \(\approx S\)

[Shamir90]: Any statement can be verified in time \(\approx\) space
Delegating Computation via Interactive Proofs

Runs in time $\text{poly}(T)$

Proof: $f(x) \equiv y$

Runs in time $\approx S$

Open: Interactive proof for any $T$-time $S$-space computation
Delegating Computation via Interactive Proofs

[Goldwasser-Kalai-Rothlum08]: Interactive proof for any $T$-time $D$-depth computation
Delegating Computation via Interactive Proofs

\[ f(x) \overset{?}{=} y \]

\[ proof: \; f(x) \overset{T}{=} y \]

\[ \text{Runs in time } poly(T) \]

\[ \text{Runs in time } \approx D \]

\[ \text{depth} \]

\[ x_1 \bar{x}_1 \quad x_2 \bar{x}_2 \quad x_3 \bar{x}_3 \quad x_4 \bar{x}_4 \]
Delegating Computation via Interactive Proofs

[Reingold-Rothblum-Rothblum17]: Interactive proof for bounded space
Delegating Computation via Interactive Proofs

Runs in time \( \text{poly}(T) \)

proof: \( f(x) \stackrel{T}{=} y \)

Runs in time \( \approx S \)

Open: Interactive proof for any \( T \)-time \( S \)-space computation
Beyond Bounded Space Computations

[Babai-Fortnow-Lund90]: Any $T$-time computation has 2-prover interactive proof where provers run in time $\text{poly}(T)$ and verifier runs in time $|x| \cdot \text{polylog}(T)$
Soundness holds only against computationally bounded cheating provers
A1 \xleftarrow{Q_1} \text{query} \xrightarrow{A_2} \text{answer}

\text{Query is independent of the computation}

\text{Computationally sound}
1. Hides $Q_1, Q_2$ from $A_1, A_2$.
2. Allows computation $A_1(Q_1)$ and $A_2(Q_2)$.

Fully Homomorphic Encryption

- [Gentry09]
- [Biel-Meyer-Wetzel99]
- [Dwork-Langberg-Naor-Nissim01]
- [Dodis-Halevi-Rothblum-Wichs16]

Not Sound!
What Went Wrong?

MIP:

\[ A_1 \text{ is a function of } Q_1 \]
\[ A_2 \text{ is a function of } Q_2 \]

This is called a local strategy.
What Went Wrong?

MIP:  
- $A_1$ is a function of $Q_1$
- $A_2$ is a function of $Q_2$

This is called a *local strategy*

In Reality

“Spooky”

Encryption:
- $A_1$ doesn’t reveal $Q_2$
- $A_2$ doesn’t reveal $Q_1$
  to poly-time observer

This is exactly *non-signaling*!

Almost...
What We Need:
Classical MIPs
Non-Signaling MIPs

Soundness holds against non-signaling strategies

Cheating provers are **not restricted** to being local
The only restriction: They **cannot transfer information**

Non-Signaling strategies [Khaln-Tsirelson 85, Rastall 85]

$A_1$ does not reveal info about $Q_2$

$A_2$ does not reveal info about $Q_1$
[Kalai-Rothblum-Raz13]:
Construct non-signaling MIP $\forall T$-time $f$, where verifier’s runtime $|x| \cdot \text{polylog}(T)$, provers runtime $\text{poly}(T)$ and communication complexity $\text{polylog}(T)$.

Queries are independent of the computation.

Sound assuming FHE is secure.
"Theorem":

Sound against non-signaling strategies

[KRR13]

Sound assuming FHE is secure

"Proof":

Suppose for contradiction there is a cheating prover, given

\[ Q_1 \quad Q_2 \]

outputs

\[ A_1 \quad A_2 \]

MIP is sound against non-signaling strategies, implies \( A_1 \) must signal information about \( Q_2 \), or \( A_2 \) must signal information about \( Q_1 \).

This breaks the security of FHE.
Bounded Space Computations
Efficient prover in some cases
From Theory to Practice

- Hyrax [WTSTW17]
- Giraffe [WJBSTWW17]
- Zebra [WHGSW16]
- Bulletproof [BBPWM18]
- Pinocchio/libSnark [PHGR13, BCGTV13]
- Pepper [SMBW12]
- Ginger [SVPBBW12]
- LibSTARK [BBHR18]
- Ligero [AHIV17]
- Buffet [WSRBW15]
- Pantry [BFRSBW13]
- Zaatar [SBVBPW13]
- Proof Carrying Data [BCTV14, CTV15]
- Scalable Zero Knowledge [BCTZ14]
THANK YOU