Lecture 17

Today: Complete IP 2 PSPACE

Few words on knowledge

Admin: No lecture on Monday - holiday

I'll be away - Swastik will lecture (1st of 3 lectures on PCP).
Review from last lecture

**Polynomial Construction Sequence**

$P_0, P_1, P_2, \ldots, P_e$

- Field $= F$
- Each polynomial of degree $\leq d$
- \# variables $\leq m$
- $P_0$ computable in time $\leq t$
- $P_i$ computable with oracle for $P_{i-1}$ with \# calls $\leq w$

$\Rightarrow$ Given $\vec{a} = (a_1, \ldots, a_m) \in F^m$ and $b \in F$

Can prove interactively that $P_e(\vec{a}) = b$

in time $\text{poly}(l, d, F, m, t, w)$. Provided $|F| \gg 1$ large.
Typical Phase of interaction

"$P_i(a^{(i)}) = b^{(i)}?$"

Verifier

**Prover**

- Compute $V_1, \ldots, V_w$
- $P_i(a^{(i)})$ can be computed from $P_{i-1}(V_1, \ldots, V_w)$.
- Compute Curve $C$ s.t. $C(j) = V_j$; $h \leftarrow P_{i-1}(C(t))$

- Verify $C(i) = V_i$
- Verify $b^{(i)} = f_i(h(1), \ldots, h(w))$

Pick $t_0 \in \mathbb{F}$ at random

$c^{(i-1)} = C(t_0); b^{(i-1)} = h(t_0)$
Poly Construction Sequence for PSPACE

**Given:** Machine $M$, $n$ s.t.
configuration of machine are
$S$ bits long.

**Goal:** To decide if initial config $a_1...a_s$
leads to (unique) accepting config $b_1...b_s$
in $2^s$ steps.

**Define:** Function

$$F_0, F_1, ..., F_s : \{0, 1\}^{2S} \rightarrow \{0, 1\}$$

$$F_i\left(\sigma = (\sigma_1...\sigma_s), T = (T_1...T_s)\right) =
\begin{cases}
1 & \text{if } \sigma \Rightarrow T \\
\text{arbitrary} & \text{for } \sigma, T \not\in \{0, 1\}^S
\end{cases}$$
1) Can define \( F_0 \) s.t. it is a polynomial of degree \( O(1) \) in each variable; and is computable in polytime.

2) \( F_i(\bar{x}, y) = \sum_{z \in \{0, 1\}^s} F_{i-1}(\bar{x}, z) \cdot F_{i-1}(z, y) \)

\( \uparrow \)

- \( F_i \) computable from \( F_{i-1} \)
- degree in each variable \( \leq C \)
- \( F_i \) needs \( 2^s \) values of \( F_{i-1} \)

\( \uparrow \)

Need to fix this
Breaking Exponential Sum into Smaller Pieces.

\[ F_{i-1} \rightarrow F_i \]
\[ G_{i,s} \rightarrow G_{i,s-1} \rightarrow \ldots \rightarrow G_{i,0} \]

\[ G_{i,s}(x, y, z) = F_{i-1}(x, z) \cdot F_{i-1}(z, y) \]

\[ G_{i,j}(x, y, z, \ldots, z_j) = G_{i,j+1}(x, y, z, \ldots, z_j, 0) + G_{i,j+1}(x, y, z, \ldots, z_j, 1). \]
The sequence

$G_{0,0}, G_{1,0}, G_{1,1}, \ldots, G_{1,0}, G_{2,0}, \ldots, G_{1,0}$

over every large field is good.

degree $\leq 4 \cdot c \cdot s$

length $\leq O(s^2)$

width $= 2$

Time $= O(s)$

# variables $\leq 3s$
(Zero) Knowledge

- Classical theory of Information [Shannon]:
  - if I send you the outcome of n unbiased coin tosses, that gives you n bits of information.

- If goal of a website is to spread information, then would have websites filled with coin tosses...

- What do intelligent entities "trade" when they exchange bits?

Claim: Want "knowledge", not "information".
Claim: Sequence of \( n \) random coin tosses has 0 bit of knowledge; (vs. \( n \) bit of information).

(More interesting) Claim: If I take primes \( P, Q \) (\( n \) bits each) and send you \( N (= P \cdot Q) \) then you don't know \( P \) (or \( Q \)).

Anecdotal Story*: Micali posed variant of above as question in problem set in U. Toronto in \( \sim 82 \); Cook responded: "I don't know how to prove I don't know!"

* Formal theory of knowledge emerged. [Goldwasser, Micali, Rackoff]
Definition by Example:

- ZK proof of Graph Isomorphism
  \[ \text{Goldreich} \]
  \[ \text{Micali} \]
  \[ \text{Wigderson} \]

Given: \( G_1, G_2 \)

Goal: Prover \( \leftrightarrow \) Verifier

- Completeness: if \( G_1 \not\cong G_2 \) then \( V \) must accept w.h.p.

- Soundness: if \( G_1 \cong G_2 \) then \( V \) must reject w.h.p.

- Zero Knowledge: if \( G_1 \cong G_2 \) then \( V \) must not know isomorphism; or learn anything other than this fact from Conv.
**Protocol**: New P Randomized!

\[ V \leftarrow G_1, G_2 \rightarrow P \]

[say $G_i = \Pi_0(G_2)$]

Pick $i \in \{1, 2\}$

$\Pi \in \nu S_n$

$H = \Pi(G_i)$

$b \in \{1, 2\}$

\[ \Pi \text{ if } b = i \]

\[ \Pi \circ \Pi_0 \text{ if } i = 1, b = 2 \]

\[ \Pi \circ \Pi_0^i \text{ if } i = 2, b = 1 \]

**Claim**: Sound

**Claim**: Zero-Knowledge
Definition of Zero-Knowledge

1. Fix verifier's coins \( R \)
2. Transcript is still random variable with distribution \( D_R \)

If Verifier can sample from \( D_R \) on its own, then Verifier gains no knowledge from prover.

"Simulator" = Sampler of \( D_R \)

[Perfect Zero Knowledge]
Simulator for GI

Verifier

\[
\begin{align*}
&\rightarrow \\
&\rightarrow \\
&\rightarrow
\end{align*}
\]

Simulator

Pick \( \pi \in S_n \)

\[
H = \pi(G_b)
\]

Output \( (H, b, \pi) \)

Exactly same distribution as with prover!

(for every \( b \in \{0, 1\} \))

Sequence important for soundness!
1. An weaken definition of Zero-Knowledge.

(i) Simulator produces

\[ D'_R \approx_e D_R \]

ie., \[ \leq \frac{1}{n} \left| D'(x) - D(x) \right| \leq 2\epsilon \]

"Statistical ZK" (SZK)

Equivalently for tests \( T: \{0,1\}^n \rightarrow \{0,1\} \)

\[ | P_{x \in D_R} [T(x) = 1] - P_{x \in D'_R} [T(x) = 1] | \leq \epsilon \]
(ii) Simulator produces $D'' \approx D_R$ s.t.

\[
\forall \text{ polytime alg. } A
\left| \Pr_{x \in D_R} [A(x) = 1] - \Pr_{x \in D''} [A(x) = 1] \right| \leq \epsilon.
\]

"Computational ZK" (CZK)

Results

- [GMR]: Definition + protocol for NP-amp. problems.

- [GMW]: GI ∈ PZK ⊆ SZK ← statistical

\[ IP \subseteq CZK \text{ if O.W.F. exist, } \uparrow \text{ cryptographic} \]
• [Fortnow, Bopanna Hästad]:
  \[ \text{SZK} \subseteq \text{co-AM} \]
  \[ \Rightarrow \text{TE} \in \text{SZK} \text{ can't be NP-hard unless } \text{PH collapses} \]

• [Okamoto]:
  \[ \text{SZK} = \text{co-SZK} \]

• [Sahai-Vadhan]; [Goldreich-S-V] etc:
  \[ \text{SZK complete problems.} \]

  \[ \text{E.g. } \text{SD} = \left\{ (C_1, C_2) \mid C_1, C_2 : \{0,1\}^n \to \{0,1\}^m \text{ circuits poly size} \right\} \]
  \[ \{ C_1 \}_{i=1}^{2} \{ C_2(x) \} \]
Easy: \( SD \leq \text{SZK} \)

Harder: \( SD \leq \overline{SD} \).

\[ (C_1, C_2) \rightarrow (D_1, D_2) \]

\[ \Rightarrow C_1 \approx^e C_2 \iff D_1 \neq^e D_2 \]

Recently, rich theory of CZK

\[ L_1 \in \text{CZK} \quad \& \quad L_2 \in \text{CZK} \]

\[ \Rightarrow L_1 \cup L_2 \in \text{CZK}. \quad [\text{Vadhan}] \]