Lecture 14: ZK Proofs on Secret-Shared Data

MIT - 6.893
Fall 2020
Henry Corrigan-Gibbs

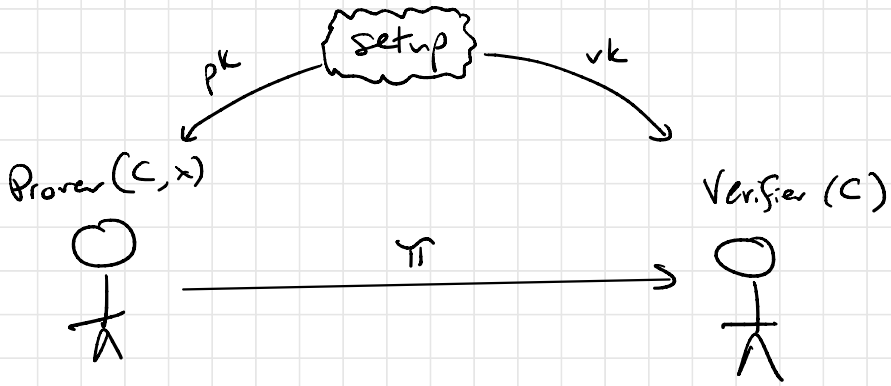
Plan

* Recap: SNARGs
* App: Private Aggregation
* ZK Proof on Secret-Shared Data
  → Defn
* Breakout rooms
* Stretch Break
* Construction of ZK $\mathcal{R}$ on Secret-Shared Data

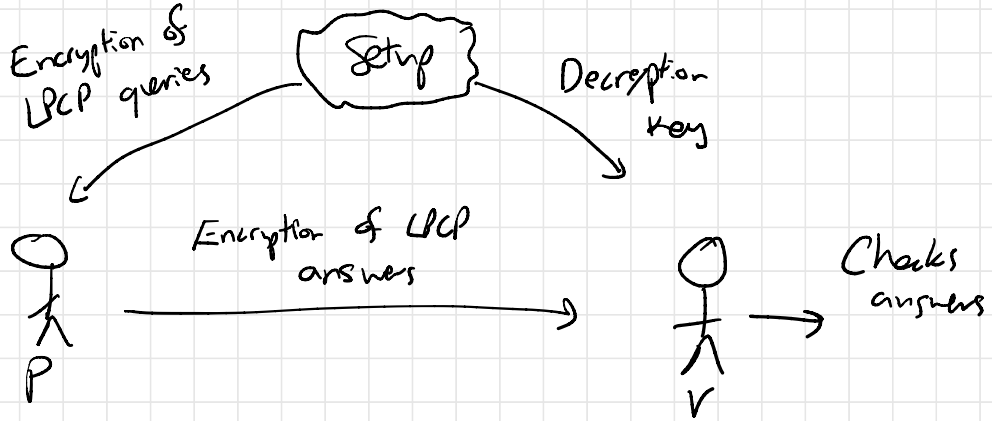
Logistics

* HW4 due Friday 5pm
* OH Today 3-4:30pm
* Guest lecture next Wednesday.

Recap: SNARG



* Prover convinces verifier that C is SAT
  (e.g. "SAT assignment starts with ____")
* Verif runs in time less than required to evaluate C
* Proof is short! Length depends only on sec param.
* Get ZK "for free"



Private Aggregation ....[Chaum '88, ...]

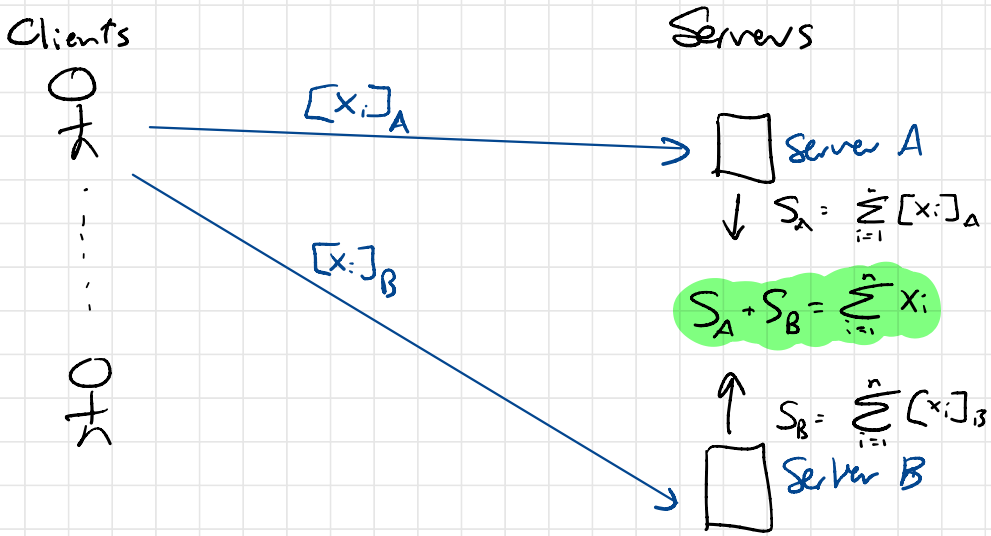
- Large # of clients holding values

$$x_1, \ldots, x_n \in \mathbb{F}$$

- Server wants to learn $\sum_{i=1}^{n} x_i \in \mathbb{F}$.

→ "Degenerate" MPC in which ckt has only addition gates — no mul gates.

Simple Protocol



Simple Protocol

1. Correctness

All parties honest $\Rightarrow$ Get correct answer

2. Privacy

≥ 1 server honest $\Rightarrow$ Protocol "leaks nothing more than $\sum_{i=1}^{n} x_i$"

If you don't care about efficiency or clients going offline, every client can act as a server.

3. Concrete efficiency

- One msg from client to each server
- One msg from servers to each other

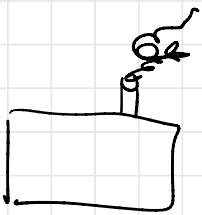
Applications

[Annoying in practice: need a second server]

Telemetry

$X_i := \begin{cases} 0 & \text{if not used Priv mod} \\ 1 & \text{o.w.} \end{cases}$

Firefox users



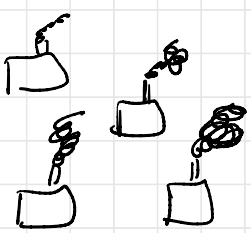
Mozilla

How many FF users used private browsing mode in last 24 hrs?

Surveys

(Boston Wage Gap)

↖ Impl by folks at BU



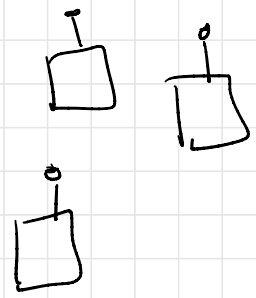
Boston employers

Civil society org

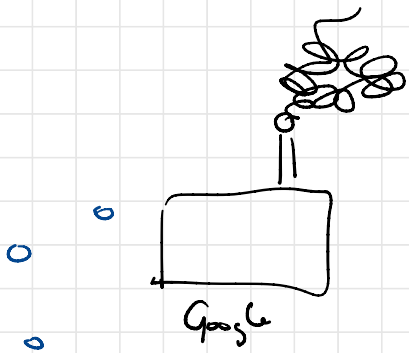
How much do M vs. W get paid on average across Boston employers?

Applications

federated learning

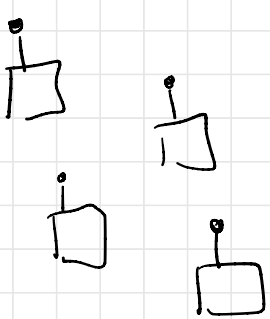


Phones



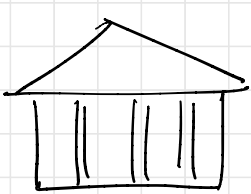
How well does our next-character prediction model work on average? (avg gradient)

Pub Health Statistics



Phones

How many miles did people travel on avg today?

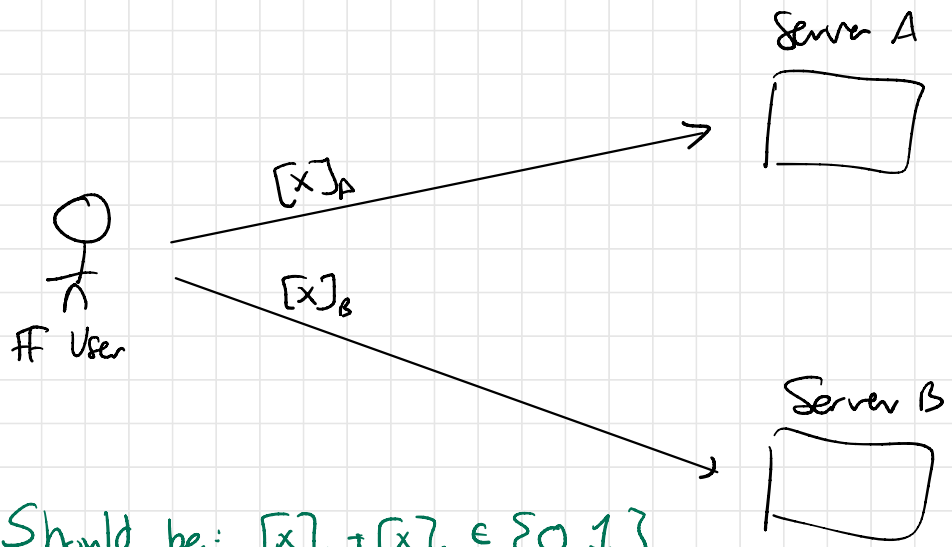


Pub health agency

Problem w/ Simple Protocol

No limits on range of client inputs.

Sum is computed $\in \mathbb{F}$ (i.e. mod p)



Should be: $[x]_A + [x]_B \in \{0, 1\}$

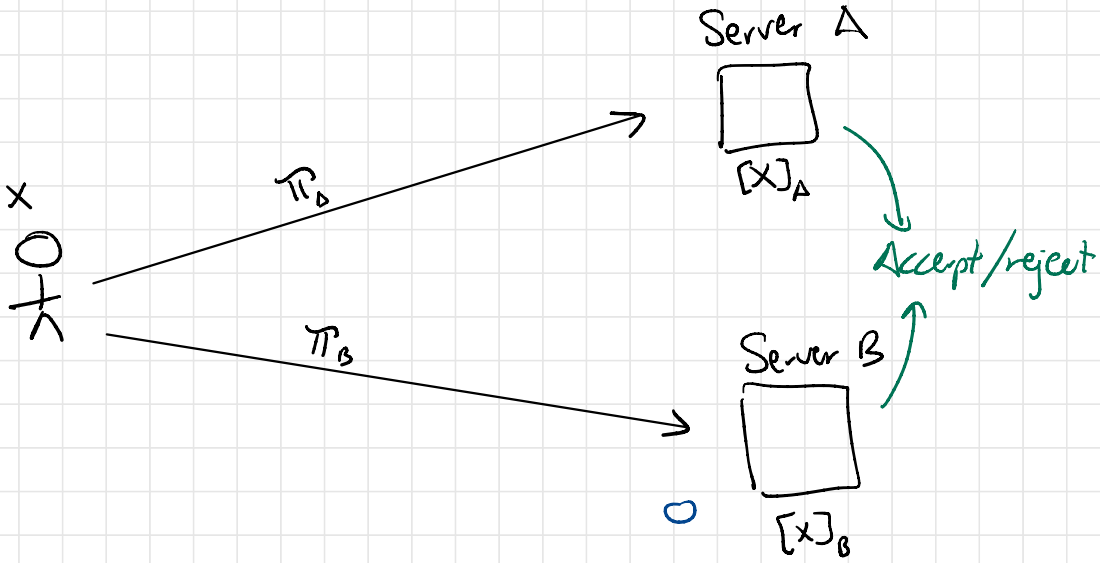
Evil user can set: $[x]_A + [x]_B =$ anything $\in \mathbb{F}$

↳ arbitrary influence on output

↳ Output is $\left(\sum_{i=1}^{n} x_i\right) + \Delta$ for adv chosen $\Delta \in \mathbb{F}$

Really, we want to compute $\sum_{i=1}^{n} x_i$ for $x_i \in \{0,1\}$

Fix using ZK

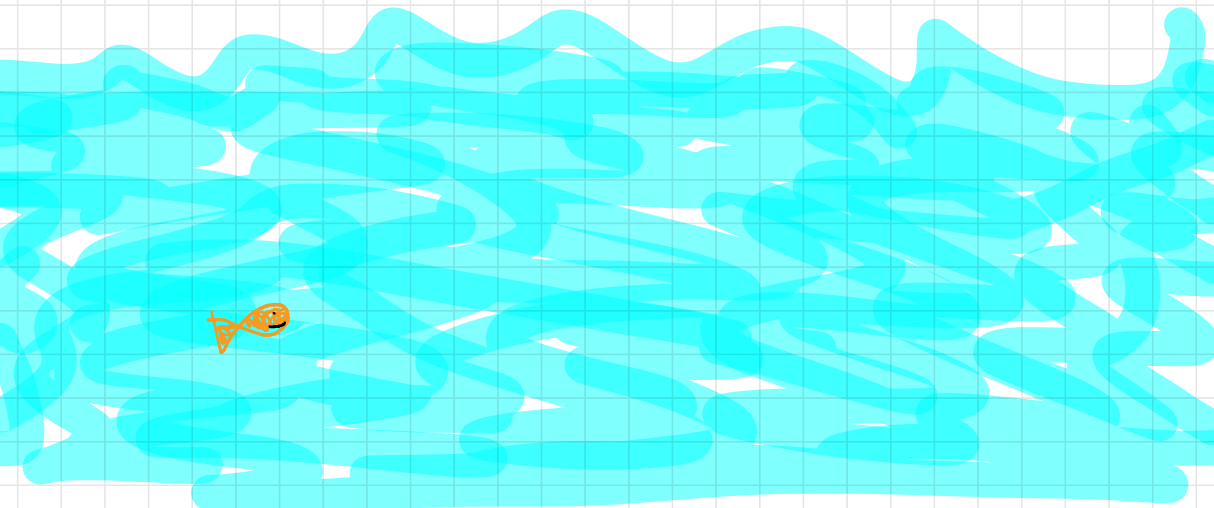


Client proves to servers that they hold shares of $x \in \{0,1\}$

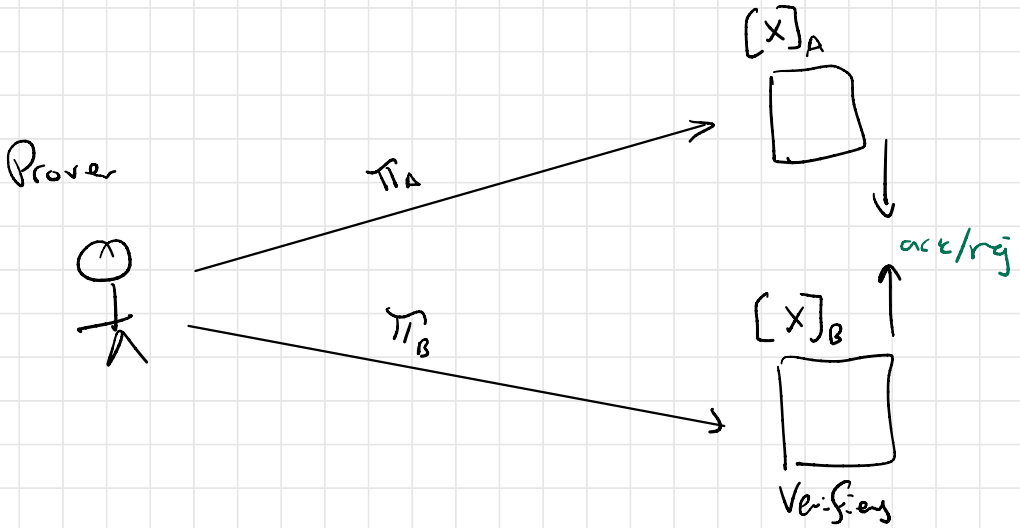
- **Complete:** Honest P (client) convinces honest Vs (servers)
- **Soundness:** Dishonest P rarely convinces honest V
- **ZK:** Servers learn nothing about x, except that $x \in \{0,1\}$
  ↳ Each server can simulate its view...

→ ZK proof on secret-shared data
Multiple verifiers, each holds a share of data.

Breakout Rooms



Constructing ZK on Secret-Shared Data

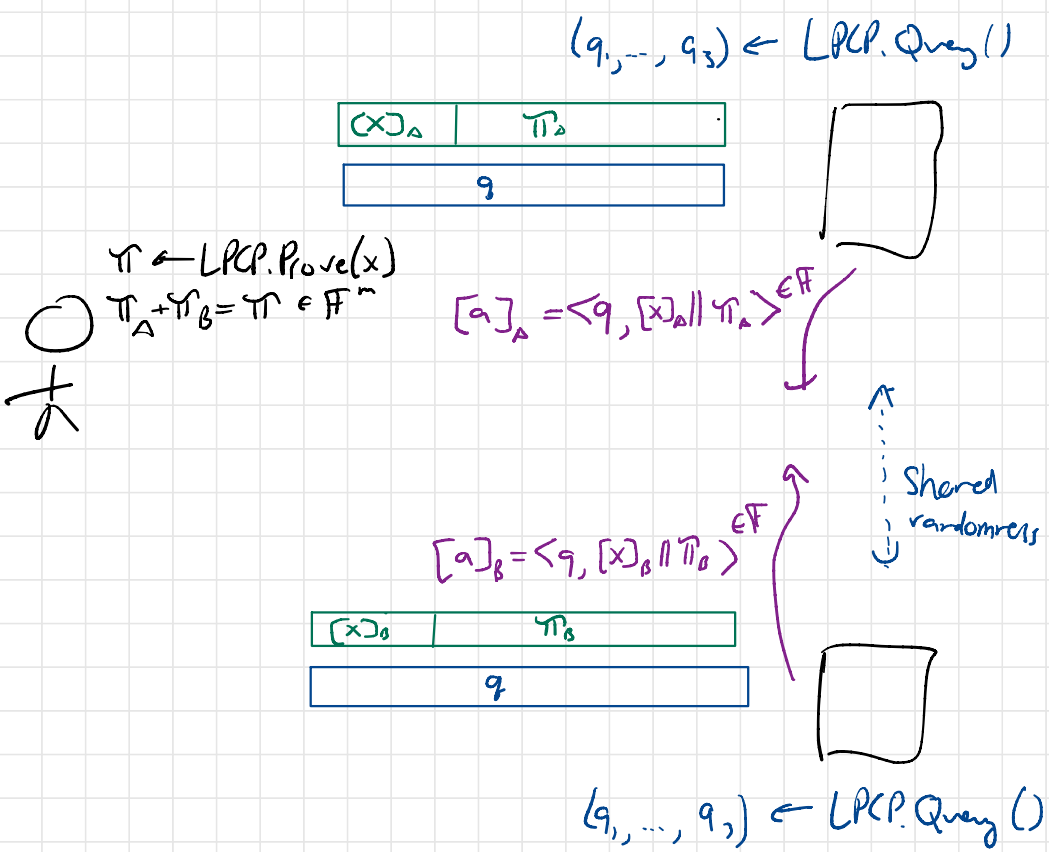


Key idea: Use linear PCPs.

$x \in \{0,1\} \subseteq \mathbb{F} \iff x(x-1) = 0 \in \mathbb{F}$

$(C(x) = x(x-1))$

1. Prover produces linear PCP proof π attesting to fact that x satisfies C.
2. Prover splits π into shares, sends one share to each verifier.
3. Verifiers jointly make linear queries to secret-shared input & proof.
   ↳ Run LPCP verif alg on answers



$$[a]_A + [a]_B = \langle q, [x]_A \| \pi_A \rangle + \langle q \| [x]_B \| \pi_B \rangle$$

$$= \langle q, x \| \pi \rangle \quad \text{as we need.}$$

As before, completeness, soundness, ZK follow almost immediately from properties of LPCP

Efficiency

$P \to V$: $O(|C|)$ $\mathbb{F}$ elms for ckt of size $|C|$

$V \to V$: $O(1)$ $\mathbb{F}$ elms ... one per linear PCP query.

Better Efficiency?

* Might like to reduce $P \to V$ communication.
* Can do if ckt C has nice structure
  e.g. many repeated sub ckts
  ↑ In some cases, requires interaction.