Lecture 13 - SNARGs

MIT - 6.893

Fall 2020

Henry Corrigan-Gibbs

Plan

* Recap: Linear PCPs
* Applications
* Stretch Break
* ~~SNARG Defn~~
* SNARG from LPCP + Lin. Hom. Enc.

Logistics

* HW4 due friday 5pm via Gradescope
* Please fill out survey on Piazza (even listeners)

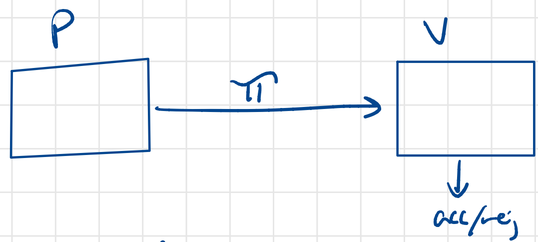
Recap: Linear PCP

(am conflating LPCP w/ Fully LPCP... distinction is not super important here)

Consider statements of form

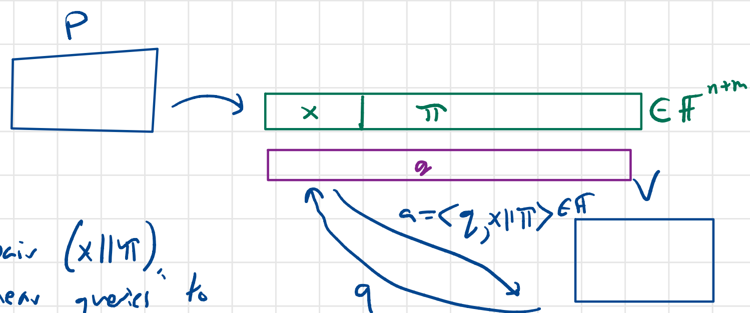
"Arithmetic ckt C is satisfiable."
(over finite field $\mathbb{F}$)

Normal proof:



* V reads entire proof
* V accepts / rejects

Linear PCP:



* P outputs pair $(x \| \pi)$.
* V makes "linear queries" to $(x \| \pi)$.
* V accepts / rejects.

Recall for, $x = (x_1, \ldots, x_n) \in \mathbb{F}^n$
$y = (y_1, \ldots, y_n) \in \mathbb{F}^n$

$$\langle x, y \rangle \stackrel{def}{=} \sum_{i=1}^{n} x_i y_i \in \mathbb{F}.$$

Q: Why does this matter?

Seems like a bogus/contrived notion... queries are as large as the proof... V can't send them.

A: These are a useful building block, as we will see today.

$$\boxed{\text{Linear PCP for ckt SAT}} \xrightarrow[\text{assumptions}]{\text{+crypto}} \boxed{\text{Succinct ZK proof for ckt SAT}}$$

[Bitansky, Chiesa, Ishai, Ostrovsky, Paneth 2013]

Properties of Linear PCP (P, V) for arith ckt SAT:

1. **Completeness**: If $C(x) = 0$, $\pi \leftarrow P(x)$

$$\Pr\left[V^{\langle \cdot, x \| \pi \rangle}() = \text{"accept"}\right] = 1.$$

2. **Soundness**: If C is UNSAT then $\forall x^*, \pi^*$

$$\Pr\left[V^{\langle \cdot, x^* \| \pi^* \rangle}() = \text{"accept"}\right] < 1/3.$$

3. **Honest verifier ZK.**

$\forall$ Sat Ckt C $\exists$ sim Sim s.t.

$$\left\{ \begin{array}{l} V\text{'s view in interaction} \\ \text{with oracle } \langle \cdot, x^*, \pi^* \rangle \end{array} \right\} \approx \left\{ Sim() \right\}$$

Constructions

We didn't look at them last time.

They're clever, but not complicated.

↳ Pretty easy to implement w/ good constants.

Thing to remember:

If C is a ckt of size s (over $\mathbb{F}$) then there's a linear PCP for C in which

* Proof has size $O(s)$
* V makes 3 queries

$(|\mathbb{F}| \gg s)$

↳ Can optimize # of queries further (Gennaro, Gentry, Parno, Raykova implicitly gave the first construction of LPCP with proof size $O(s)$ ... 2013)

Intuition

Why the existence of constant-query LPCPs should surprise and delight you.

* Normal proof: If satisfying input is n elements long, V has to read all n elements to check proof.

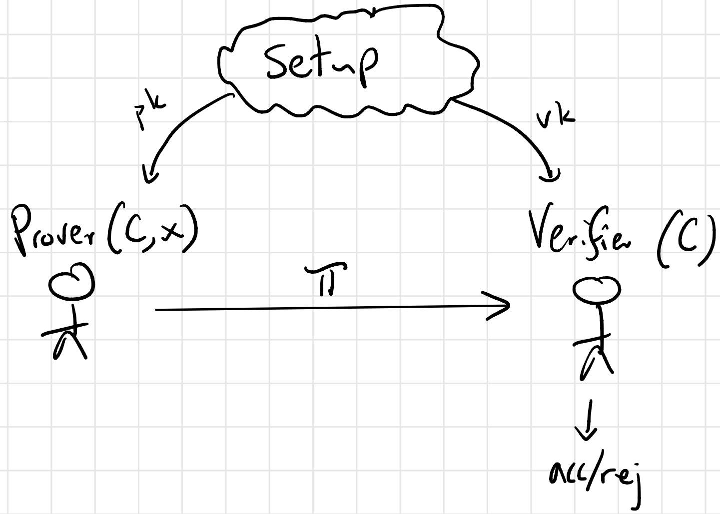
* Linear PCP: Verifier gets only a constant # of field elms worth of info back from the proof.

No matter how big ckt is or how long SAT assignment is!

...until you see it, it's hard to understand how this could be possible.

ZK

Succinct Non-interactive Argument (SNARG)

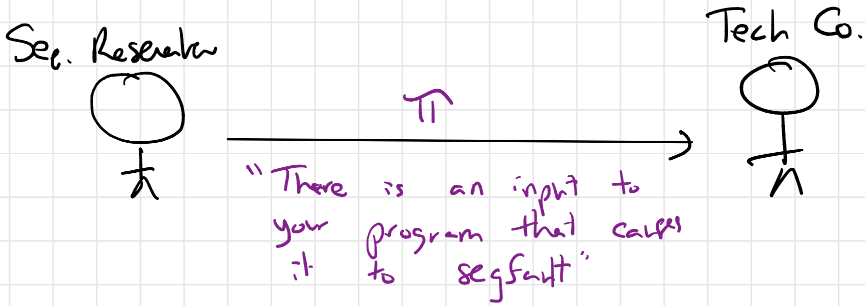


* Short proof that convinces V that C is SAT
  ↳ complete, sound
* Zero knowledge: Leaks no other info (simulation)
* Succinct: → |π| depends only on sec param...
  not on size of circuit or sat assignment
  → Time to check proof also depends only on sec param

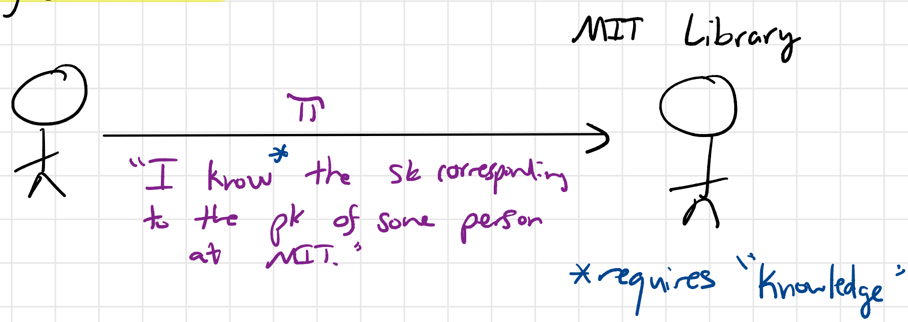
Notice: → π could be much smaller than the NP witness (sat input).
→ Useful property even w/o ZK!

Applications

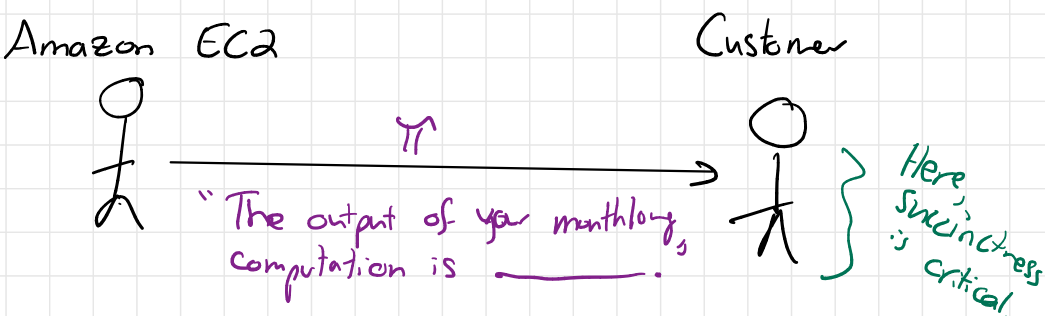
Zk Bug Bounty



Anonymous Auth



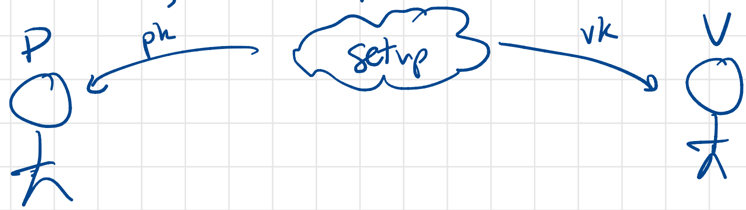
Delegated computation



Stretch Break!

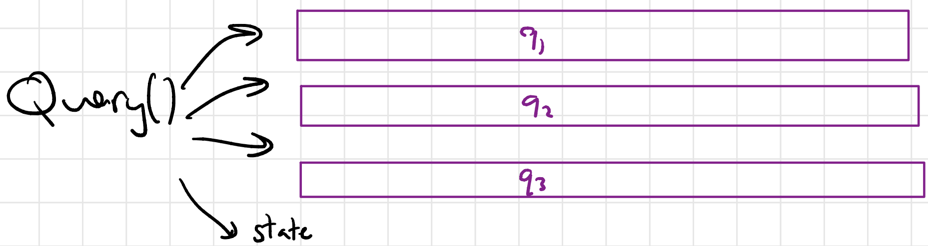
Constructing SNARGs from Linear PCPs

As a simplifying assumption, let's first consider "designated-verifier SNARGs":



Soundness only holds if prover cannot get ahold of secret vk

Furthermore, assume LPCP verifier has structure



$$\mathrm{Decide}(\mathrm{state}, a_1, a_2, a_3) \to \mathrm{acc/reject}$$

In other words, the verifier's queries are non-adaptive and are indep of the statement being proved.
circuit

Construction

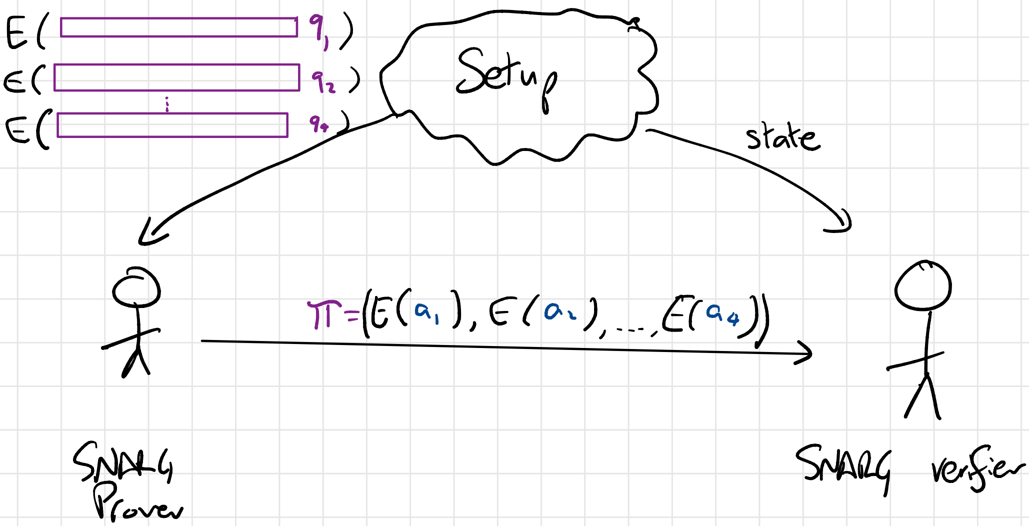
Uses linearly homomorphic encryption w/ keyspace $\mathcal{K}$

$$E(sk, m_1) + E(sk, m_2) = E(sk, m_1 + m_2)$$

... can build from an array of "public-key assumptions"

Paillier, DDH, LWE, ...

If ckt & LPCP over field $\mathbb{F}$, then msgs in lin hom enc scheme should be $\mathbb{F}$ elements.



- Proving key = Enc of LPCP queries
  Verif key = LPCP verif state
- SNARG prover computes LPCP answers "under encryption"
- SNARG verfier decrypts and runs LPCP verifier

Setup( ):

$(q_1, q_2, q_3, state) \leftarrow$ LPCP.Query()

Choose random $\alpha_1, \alpha_2, \alpha_3 \xleftarrow{R} \mathbb{F}$

$q_4 \leftarrow \sum_{i=1}^{3} \alpha_i q_i \quad \in \mathbb{F}^{n+m}$

$sk \xleftarrow{R} \mathcal{K}$

return $pk = (E(sk, q_1), \ldots, E(sk, q_4))$

$vk = (state, \alpha_1, \alpha_2, \alpha_3, sk)$

Prove$(pk = (Q_1, \ldots, Q_4), x)$:

$\pi \leftarrow$ LPCP.Prove(x)

For $i = 1, \ldots, 4$:

$A_i \leftarrow \langle Q_i, \pi \rangle$

return $(A_1, \ldots, A_4)$

Verify$(vk = (state, \alpha_1, \ldots, \alpha_4, sk), \pi = (A_1, \ldots, A_4))$:

For $i = 1, \ldots, 4$: $a_i \leftarrow Dec(sk, A_i)$

Reject if LPCP.Verify$(state, a_1, \ldots, a_4)$ = "reject"

Reject if $a_4 \neq \sum_{i=1}^{3} \alpha_i a_i \quad \in \mathbb{F}$

Accept!

This is a very slick construction!

.... No craziness hiding. It's really clean and even easy to memorize.

If you're stuck on a desert island and need a succinct proof system, this is what you'd use.

Soundness:

- Essentially follows from LPCP soundness.
- Only tricky part is that P^* can answer diff queries w/ diff proofs
  - ↳ Random linear comb defeats this attack

→ Need a new assumption "linear-only enc" to formally argue soundness. Not great, but also no reason to suspect these assumptions are more false than any other crazy assumption we make.

Zk:

- Verifier only gets answers to LPCP queries (computed honestly in setup)
- Zk of SNARG follows directly from Zk of LPCP.

Q: Can P & V reuse setup for multiple interactions?

A: Yes, prove statements of the form "$C(x)$ is SAT and first ℓ elms of a SAT input are ......".